椭圆曲线密码体系研究

· 肖攸安 著 ·

華中科技大學出版社
http://www.hustp.com

ISBN 7-5609-3858-2

定价：15.00 元

椭圆曲线密码体系研究

肖攸安 著

华中科技大学出版社

图书在版编目(CIP)数据

椭圆曲线密码体系研究/肖攸安 著
武汉:华中科技大学出版社,2006年10月
ISBN 7-5609-3858-2

Ⅰ.椭…
Ⅱ.肖…
Ⅲ.因特网-安全技术
Ⅳ.TP393.4

椭圆曲线密码体系研究 肖攸安 著

责任编辑:余 涛 封面设计:潘 群
责任校对:吴 晗 责任监印:张正林

出版发行:华中科技大学出版社
武昌喻家山 邮编:430074 电话:(027)87557437

印 刷:华中科技大学印刷厂

开本:850×1168 1/32 印张:8.125 字数:220 000
版次:2006年10月第1版 印次:2006年10月第1次印刷 定价:15.00元
ISBN 7-5609-3858-2/TP·621

内 容 提 要

椭圆曲线密码体系是当前信息安全领域的研究热点之一，本书在分析和研究椭圆曲线密码学的最新研究成果的基础上，分7章总结了作者在该领域所完成的一系列的研究工作。其中，第1章从网络信息安全现状出发，分析了所面临的安全威胁，归纳了人们所提出的安全需求，给出了相应的解决方案，引出了椭圆曲线公钥密码体系。第2章主要介绍和讨论了在本书中所要用到的椭圆曲线密码体系的基本数学理论基础和相关的背景知识。第3章在介绍有限域上的离散椭圆曲线的基础上，深入讨论了椭圆曲线有限群上的椭圆曲线离散对数问题，归纳了安全椭圆曲线选取准则。第4章研究了椭圆曲线有限群阶的计算问题，深入研究了SEA点算法。第5章根据安全通信的需要，在讨论通信协议安全性问题的基础上，研究和分析了作者所设计的可用于椭圆曲线密码体系的密钥生成、密钥协商、密钥分配、信息加密、数字签名等多种安全高效的密码方案。第6章和第7章深入研究了椭圆曲线密码体系实现中的若干关键技术，给出了典型方案的具体实现算法和实验结果。

本书适用于信息、计算机及相关专业的博士、硕士研究生和高年级本科生，也可作为信息安全领域的研究人员和专业技术人员的参考书。

前　言

随着信息技术和网络技术的飞速发展，信息安全问题成为人们日益关注的焦点。作为信息安全技术重要基础的密码学，其主要目的是防止信息系统内的机密信息被非法访问者破译，使机密信息和重要数据得以不必通过专用的特别设施进行传输和储存，大大降低信息传输的成本和信息存储的费用。

公钥密码学是密码学的重要分支，它基于某一类公认的、在计算上不可行的数学难题，使用一对不同的密钥完成信息保密任务。由于公钥密码体系所用到的一对密钥中的一个密钥是可以公开的，因此它可以被广泛应用于诸如数字签名、身份认证、数据加密和密钥管理等众多领域，对信息安全技术具有非常重要的意义。

椭圆曲线离散对数问题是目前公认的可用于公钥密码体系的三大数学难题之一。基于椭圆曲线离散对数问题而构筑的公钥密码体系就是椭圆曲线公钥密码体系。与其他公钥密码体系相比，椭圆曲线公钥密码体系具有密钥短、强度高、参数少等特殊的优势，特别适用于空间受限、带宽受限等场合，因此得到了人们的广泛关注。经过10余年的研究，椭圆曲线公钥密码体系开始从学术理论研究阶段逐步走向实际应用阶段，成为目前最有前途的一种公钥密码体系，极有可能成为现存公钥密码体系的替代者。因此，加速对作为信息安全技术的核心基础之一的椭圆曲线密码学的研究，对于促进我国信息化工程建设的高速发展，增强我国的经济竞争实力，维护我国的主权独立和战略安全，具有十分重要的意义。

目前，椭圆曲线公钥密码体系开始从学术理论研究阶段走向

应用实现阶段，受到学术界、开发商、政府部门、密码标准研制组织等有关各界的重视，成为当前密码学界的研究热点，是现今最有前途的公钥密码体系。

虽然目前国际上已对椭圆曲线公钥密码体系进行了较为广泛的研究，但在国内，尚处于起步阶段，与国际先进水平相差较远。本书针对实际应用的需要，对椭圆曲线公钥密码体系及其相关的理论及应用技术展开了深入的研究。

具体而言，本书对椭圆曲线公钥密码体系研究的贡献主要体现在以下几个方面。

（1）*在密钥生成方面*

提出和实现了三种高效快捷的、适用于不同场合的真随机密钥生成方法XRNGS，并设计了相关装置；设计和实现了新型基本密钥对生成算法；研究了公钥可信度问题，并申请了国家发明专利（03128073.0，200510018917.5，200510018918.X，200510018920.7）。

（2）*在密钥管理领域*

设计和实现了新型高效的XKAS密钥协商方案和XKDS密钥分配方案，针对不同条件下的应用问题，研究了相应的改进和扩展方法技术。以此为基础，设计并实现了XKAS密钥协商方案，并申请了两项国家发明专利（03128072.2.公开号：CN1455543A；申请号：03128074.9.）。

（3）*在数据加密方面*

对原有的EC-ElGamal加密算法进行了改进，提出了XEC-ElGamal数据加密算法，设计和实现了基于密钥共享思想的混合数据加密方案XHES，该方案结合了两种密码体系的优点，具有很好的性能和实用价值，并申请了国家发明专利（03128222.9）。

目　录

第1章 绪 论

随着计算机网络特别是因特网的迅猛发展，数字化社会基本成型。网络的开放性使得运行在网上的各种商务活动、政务活动等网络通信活动的安全问题显得更为突出。可以说，安全问题是一切基于网络的通信活动得以正常运行的前提和基础。为了更好地研究网络信息安全问题，本书在介绍网络经济和电子商务发展以及网络信息安全现状之后，通过分析网络通信活动所面临的各种安全威胁、总结人们所提出的各种安全需求、研究相应的安全策略和相关的安全技术，引出了本书的研究主题——椭圆曲线公钥密码体系。

1.1 网络信息安全

Internet 和计算机网络的飞速发展，社会信息化步伐的加快以及网络通信的国际化、信息化、无纸化、低成本、高效率等，使得基于网络的电子商务、电子政务等各种网络社会活动受到了全世界的广泛关注，得到了极其迅猛的发展。

据统计，由于网络技术的导入，美国从1995年到1999年的秘书数量减少了17%，批发和零售业的采购人员减少了16%，节约了大量人力成本。2000年全球网上交易总额达1970亿美元，2001年则达到了3810亿美元，而2002年突破8000亿美元大关，2004年整

体营业额更是达到了创纪录的27748亿美元，几乎成几何级数地增加。通过电子商务实现的交易已经占全球贸易总交易额的20%以上，拥有相当的分量。

在我国，基于网络的各种商务和政务活动也获得了长足的发展。2005年版的《中国电子商务盈利模式研究报告》等资料的数据显示：2004年，我国已建成各类配送中心1000多家，网上银行50余家，企业与个人客户超过1000万户。2004年，我国电子商务的增长率为73.7%，营业额达到4800亿元人民币，网上购物在线支付交易总金额达到6.8亿元，2005年达到15.7亿元。截止到2005年4月，国内互联网用户人数已经超过1亿，其中参与网上购物的比例为37.9%。这说明网络经济、电子商务等基于网络的社会经济活动正在得到飞速发展，它们将成为未来信息社会经济发展的主要推动力。

虽然网络经济中孕育着巨大的商机和财富，吸引了大量投资，但也吸引了罪犯的注意力，网络犯罪的比例日益增加。

在过去的几年中，出现了一连串针对网络，特别是针对因特网的攻击。这些攻击影响巨大，不仅造成了巨大的损失，而且还严重打击了人们对电子商务的信心，使得人们几乎“谈网色变”。

在1999年3月爆发的Melissa病毒和2000年5月爆发的LoveLetter病毒都是利用Outlook电子邮件附件进行传播的，另外恶意代码也都是利用Microsoft公司开发的Script语言缺陷进行攻击的，所不同的是Melissa是Microsoft Word宏病毒，而LoveLetter则是VBScript病毒。Outlook的用户数量众多，使得这两种病毒能够迅速蔓延并造成了极大的危害。它们引起了当时人们对信息安全现状的深思，无形中对信息安全的设施和人才队伍的发展起了很大的促进作用，刺激了企业和公司对网络安全的投资，使得专业的网络安全紧急响应小组得以出现和壮大。

2000年2月，人们刚刚为基本解决“千年虫问题”而松了一口气时，又迎来了分布式拒绝服务攻击DDoS的闪电般突然袭击：全球知名网站雅虎第一个宣告因为遭受分布式拒绝服务攻击而彻底崩溃后，紧接着Amazon.com、CNN、E*Trade、ZDNet、Buy.com、Excite和eBay等其他7大知名网站也几乎在同一时间彻底崩溃。虽然，这场危机仅持续了10个多小时，但其影响却十分深远。因特网上大量的机器进行分布式计算，如DDoS、分布式扫描和分布式口令破解等，使得一个攻击者能够达到许多意想不到的强大效果，并直接导致了2002年的针对因特网寻址系统DNS的主要根服务器的DDoS攻击，几乎导致整个Internet崩溃。

2001年7月出现的基于微软IIS缓冲溢出漏洞进行感染的红色代码变种蠕虫病毒，在首次爆发的短短9个小时内，以迅雷不及掩耳之势感染了250 000台服务器，其速度和深入范围之广引起了全球媒体的注意。红色代码蠕虫不仅篡改英文站点、发动DoS拒绝服务攻击、格式化目标系统硬盘，还在每月20日～28日对白宫的WWW站点的IP地址发动DoS攻击，迫使白宫的所有WWW主机都不得不全部更改自己的IP地址。之后，红色代码不断出现变种，其破坏力也更强。在红色代码II肆虐时，有近2万台服务器和500万个网站被感染，造成了更加惊人的损失。

2001年，“9·11”恐怖袭击事件发生一个星期后出现的尼姆达Nidma蠕虫病毒，利用了IIS缺陷、IE浏览器和Outlook的JavaScript脚本执行的缺陷、硬盘共享的缺陷等至少4种微软产品的漏洞，通过多种不同的途径来进行传播，能感染多种Windows操作系统，仅用了不到半小时就传遍了整个世界，在全球各地攻击了830万台计算机，占用了大部分的网络带宽。Nidma先后出现了9代变种，累计造成将近10亿美元的经济损失。

2003年出现的SQL Slammer蠕虫王和最近出现的Blaster冲

击波病毒，都是利用系统漏洞感染了近70%的Windows网络，从而导致因特网大面积堵塞，使整个网络面临全面瘫痪，造成了巨大的经济损失。特别是冲击波病毒及其变种，不仅感染了近百万台计算机，而且还被认为是"8·14"北美大规模长时间停电事件的罪魁祸首，造成了不下500亿美元的直接经济损失。

除了病毒攻击和恶意攻击以外，直接针对网络社会的政治经济活动的攻击也在不断增加。除了因特网网站以外，卫星通信、ATM和无线网络通信也成为新的攻击目标。

2002年，"资料隐码"自动攻击程序利用大型主机Unix系统的溢出漏洞以及SQL Injection方法，通过网站查询参数，将攻击代码植入网站的数据库，穿透防火墙，直接盗取电子商务和网络银行数据库中的个人资料和密码，骗过交易安全审核机制，展开非法网络交易。据报道，近80%的网络银行、70%的电子商务网站都沦为该程序的攻击目标。欧、美、韩、日等连续发生大规模的身份盗用和冒领欺诈事件，黑客轻易地接管账户，进行虚假交易，让不知情的合法使用者买单付费。一时间人心惶惶，网络电子交易活动陷入低谷。仅仅在2003年2月，就有多达220万个维萨(VISA)和万事达(Master)卡账户、3500万个AOL账号的关键信息被盗，迫使多家银行不得不暂时关闭相关业务。

RSA公司在2003年6月发布的统计数据称：仅2002年，就出现了62 000起黑客攻击事件，因数据被盗窃、身份证被盗窃等事件引发的损失就高达590亿美元。

Card Systems公司主要为万事达、维萨等公司提供信用卡服务，每年能处理高达150亿美元的信用卡客户和商户的结账。2005年6月，恶意黑客闯入了Card Systems公司的客户资料库，窃取了多达4 000万个信用卡账户的资料，导致数十万名用户的账户被盗用，成为有史以来最严重的信息安全事件之一。

美国联邦调查局和美国互联网欺诈投诉中心在2005年8月发布的统计数据显示:2005年上半年,共发生2.37亿次安全攻击活动,同比增长50%,针对政府、金融服务、制造厂商和医疗部门的事件日益增加。我国国家计算机网络应急技术处理协调中心的统计数据表明:2004年针对金融网站和电子商务网站的网络仿冒诈骗攻击比2003年增长了220多倍,中国银行网站、中国工商银行网站等多家金融网站均遭到这类攻击,造成了巨大的经济损失。

所有这些都大大减弱了人们对网络经济的热情,减慢了电子商务的发展速度。因此,如何保障维护网络信息社会的正常秩序、保障网络信息的安全,成为当前急需解决的首要问题。

全球领先的网络安全技术公司赛门铁克公司在2005年9月发布的2005年度上半年互联网安全威胁研究报告中指出,中国受到的网络攻击数量占全球检测到的攻击总数的6%,仅次于美国,位列第二;此外,有18%的攻击事件和攻击跳板来自中国。而另一方面,我国的信息安全技术水平目前还处在低水平状态,在世界范围内,被排在等级最低的"第四类"。因此,我们必须努力促进信息安全技术领域的自主开发,涌现一批具有自主知识产权的信息安全产品。对网络信息安全的研究不仅是必要的,而且是急需的。它不仅关系到我国网络经济和信息化建设的成败,更关系到国家利益和国家安全。

1.2 安全威胁和安全需求

对网络信息安全研究而言,首先需要明确网络社会所受到的各种安全威胁,归纳出人们对网络安全性的一系列需求,为寻求进一步的解决策略做准备。本节将针对网络信息安全的研究需要,首

先分析网络信息社会所受到的安全威胁以及人们提出的安全需求。

在本书的讨论中，始终假定通信各方是通过一条不安全的、开放的公开信道（如 Internet）相互连接的，并认为恶意的第三方（以下统称攻击者）拥有强大的计算能力，有能力控制网络上的信息流的传输。这样，在通信各方之间传输的信息就有可能被攻击者非法阅读和篡改。这里需要注意的是，攻击者并不一定都是局外人，有可能就是系统的一个或多个合法用户。在某些情况下，甚至可能就是通信的一方（如电子交易中的欺诈者）。

对于一个网络通信系统，可以根据信息流的流动状况来分析其工作是否正常、是否受到攻击。

在正常情况下，信息流是从数据发送方的数据源流到数据接收方的目的地，这种正常情况下的信息流动可以用图 1-1 来表示。

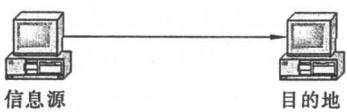

图 1-1　正常的信息流动

当信息流动与图 1-1 所描述的不同时，说明通信过程受到了攻击。这些攻击可以按不同的规则和标准进行分类。在本书中，作者参考了 Steve Kent 提出的攻击分类法，结合目前的网络信息安全的发展情况，根据攻击过程中攻击者的介入程度将这些攻击分成被动攻击和主动攻击两大类。

1.2.1　被动攻击

被动攻击（Passive Attack）是指攻击者在不干扰信息流动的情况下，从网络通信双方所交换的信息数据流中获取所需要的信

息。被动攻击在实践中表现为窃听和监视。由于它并不干扰消息的传输，所以非常难以检测，但可以防止攻击者从所得到的信息流中获得有效的信息。

一般地，被动攻击下的信息流动可以用图1-2来表示。

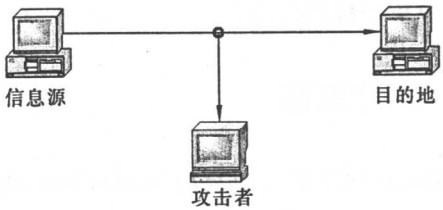

图1-2 被动攻击

根据对所截获的信息的处理方式不同，被动攻击又可以细分为两种类型：析出消息型和通信量分析型。

1. 析出消息

析出消息是指直接从信息数据流中获取所需要的消息，例如，攻击者截获了正在传输的电子邮件并获知其中的具体内容。

2. 通信量分析

当通信的双方使用加密等技术对所传输的消息进行了屏蔽和变换，使得攻击者无法直接从所截获的信息中析出消息时，攻击者可以通过观察所交换的信息的频率、长度等通信量，应用统计学方法分析消息的性质，猜测消息中所包含的可能的内容，这就是通信量分析。

由于被动攻击难以被检测，所以，对付被动攻击的重点在于防范，即防范攻击者从所截获的信息流中获得有效的信息。

对于析出消息型被动攻击，可以用加密技术来预防；而对于通信量分析型被动攻击，则要求经过加密等处理后的信息在统计学上无规律，即该加密算法具有雪崩效应，且每次传输过程中，相同的消息经过加密处理后所得到的信息不同——通过一次一密来预防这种通信量分析型被动攻击。

1.2.2　主动攻击

对于网络通信活动而言，主动攻击(Active Attack)的危害更大，特别是通信各方彼此互不信赖时，这种攻击对通信活动的威胁就显得更为严重。与被动攻击相反，主动攻击者直接参与网络通信，干扰信息的流动，篡改信息的内容，甚至产生虚假的信息流。具体地说，主动攻击有四种类型：伪装攻击、重放攻击、篡改消息和拒绝服务。

1. 伪装攻击

伪装攻击是指攻击者通过产生一个虚假的信息流，伪装成另一个合法的通信实体，参与通信过程。伪装攻击一般与其他类型的攻击联合使用。例如，攻击者可以截获过去的某一合法的鉴别信息，通过对该信息的重放，伪装成某一合法的通信实体参与通信。

伪装攻击下的信息流动如图1-3所示。

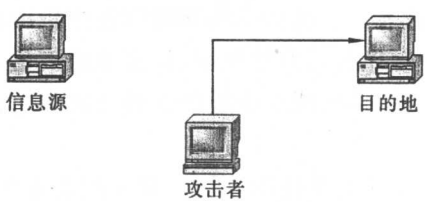

图1-3　伪装攻击

2. 重放攻击

重放攻击是指攻击者通过重传由被动攻击截获某一通信信息流的方式，参与通信过程，以产生一个未经授权的效果。重放攻击一般与其他类型的攻击联合使用，其信息流动如图1-4所示。

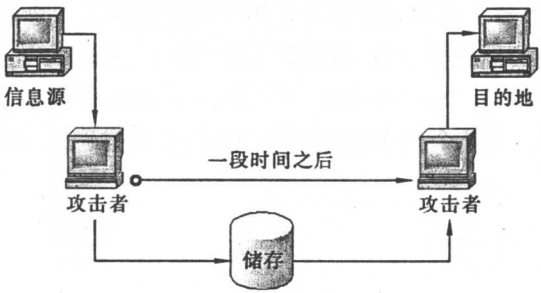

图1-4 重放攻击

3. 篡改消息

篡改消息是指攻击者通过篡改合法的通信信息流的某一部分，或者改变消息的传输顺序，甚至使消息的某一部分丢失，以求得到一个攻击者所期望的结果。例如，攻击者通过篡改消息，改变电子商务中的交易金额而获利，用图1-5可描述这一类攻击。

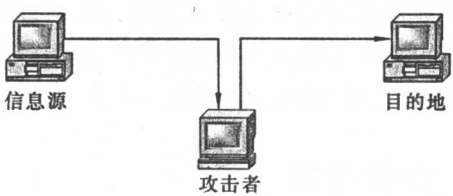

图1-5 篡改攻击

4. 拒绝服务

拒绝服务是指攻击者通过干扰信息通道，中断正在进行的网络通信或降低网络通信信道的性能，破坏网络通信系统的可用性。这种攻击一般有一种确定的特殊目标和目的。例如，在最近针对白宫邮件服务器主机的拒绝服务攻击就是通过滥发大量的反战抗议邮件，使得目标主机过载，降低其工作性能，直至最后停止工作。而红色代码等病毒针对Windows NT、SQL Server 等类型主机发起拒绝服务攻击，使整个网络陷于瘫痪，用图 1-6 可描述这一类攻击。

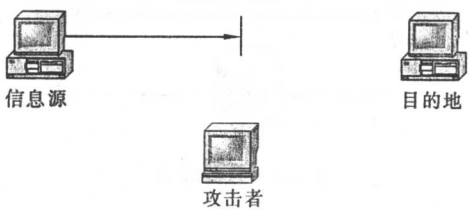

图 1-6　拒绝服务攻击

被动攻击虽然难以被检测，但可以采取适当的措施来防范。而主动攻击恰恰相反，要想完全防范主动攻击需要整个网络通信信道在任何时间里都能够得到完整的物理保护，而这在实践中是不可能的。所以，应对主动攻击的重点在于及时地检测到攻击者发起的主动攻击，并能够迅速地从主动攻击所引起的破坏中恢复——灾难恢复。检测的结果还应该具有法律效力，使得针对主动攻击的检测成为一种威慑力量，以减少主动攻击的发生。

1.2.3　安全需求

通信过程中所传输的信息可能关系到个人、企业或国家的机

密，然而人们又希望在没有专门的保密通道情况下，能利用开放的公开信道传输信息。为了能够在这样一个环境下完成安全的、保密的通信，人们对安全技术提出了各种各样的要求。总的来说，根据上一节介绍的安全攻击分类法所归纳的安全威胁，所有这些有关通信和信息传输过程中的各种安全需求可以进一步地被归纳为下面四种类型的需求。

1. 保密性需求

在传统的通信过程中，可以用邮寄封装的信件或通过可信渠道发送的商业报文来达到保守机密的目的。在开放的网络环境中，预防信息在传输过程中被非法窃取则成为人们的首要需求。

保密性是指在没有专门的保密通道的情况下，保证在通信过程中通信各方之间的通信内容不被未授权的第三方所阅读。它可以保护在开放的通信信道中传输的信息免受被动攻击，防止攻击者析出消息内容，保护通信量免受分析。在实际应用中，可以根据信息的机密程度确定合适的保密范围。

2. 真实性需求

真实性需求是指通信的一方能够鉴别另一方(数据发送方/数据接收方)的身份，确保数据源的可靠性和真实性(没有被伪造)，确定真实的数据来源。它关心的是参与通信的实体是否真正是他们所宣称的那个实体，即该通信是可信的。在必要的时候，当通信的双方关于所传输数据来源的真伪发生争执时，可以由合法的、公正的第三方来解决双方的争执。这一需求也被称为单向认证(One-way Authentication)需求。

3. 完整性需求

与真实性需求不同，完整性需求是针对数据的有效性而提出来的，它与数据产生后是否被修改过密切相关。具体地说，完整性要求防止未经授权的数据被修改，保证信息在传输过程中没有冗余、插入、篡改、重排、伪造或者丢失。完整性和保密性的关系很紧密，破坏了完整性也就意味着破坏了保密性，因为能改变信息的窃听者肯定能阅读此信息。完整性和保密性之间的差别在于：对保密性的安全威胁是指未经授权的人看到了不应该看到的信息，而对完整性的安全威胁则是指某人对信息的内容做了未经授权的改动。

在电子商务活动中，这一需求尤为重要。数据传输过程中信息的丢失、信息重复或信息传输的次序差异以及可能的欺诈行为或其他意外差错，都会导致贸易各方所得到的信息不同。贸易各方信息的完整性将影响到贸易各方的交易和经营策略，所以，保持贸易各方信息的完整性，防止信息被随意生成、修改和删除，同时防止数据传输过程中信息的丢失和重复并保证信息传输次序的统一是电子商务应用的基础。

4. 不可否认性需求

不可否认性也称抗否认性、不可抵赖性等。由于通信过程中参与通信的各方不是面对面地会谈，所以，人们期望能够有一种技术能保证通信各方对自己行为的承诺，防止人们否认自己曾经做过的行为。具体地说，不可否认性需求一方面要求数据的发送方在发送出消息之后，就再不能否认他所发出的消息；另一方面，要求数据的接收方在收到消息后不仅能证实消息确实是由其所宣称的那个发送者所发送的，而且事后也不能否认曾经接收过该消息。

1.3 解决方案

针对网络通信过程中所面临的安全威胁和1.2.3节所归纳的各种安全需求，Dolev和Yao提出了如图1-7所描述的网络安全模型。

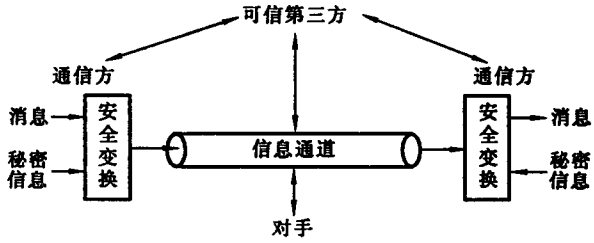

图1-7 网络安全模型

这一网络安全模型将所有的安全技术分为3个部分。

① 对将被发送的信息进行安全性相关的变换，包括消息的加密，加密可搅乱该消息，使对方不可读；增加基于该消息内容的新代码，使之能够证实发送者的身份。

② 收发双方各自享有一些不为对方所知的秘密信息，用于控制在连接前对消息的搅乱变换以及接收后对消息的恢复变换。

③ 为保证安全传输，解决通信中可能出现的争端，需要一个被通信的双方信赖的、权威的、可信的第三方，负责仲裁、解决有关通信收发双方对消息传输的真实性的争议，或者负责向通信收发双方分发密钥等秘密信息。

因而，在设计网络通信安全系统时有四个基本任务。

① 设计一个算法来执行安全性相关的变换，使对手不能得到

变换后的信息。

② 生成用于该安全性变换算法的秘密信息。

③ 研究秘密信息的分布和共享的方法。

④ 规定由通信双方使用的协议，该协议利用安全变换算法和秘密信息以获得保证网络通信安全的服务。

为了完成上述四个基本任务，Whitfield Diffie 和 Martin Hellman 在 1975 年提出了基于“陷门单向函数(Troopdoor One-way Function)”思想的三种解决方案的思路，并讨论了这三种方案之间的关系，开创了公钥密码学这一密码学研究的新方向。之后，人们进一步将这三种解决方案发展成如下三种形式的公钥密码体系。

1. 公钥加密体系

在传统的对称加密体系中，加密过程和解密过程是互逆的，二者采用一个共同的密钥。在公钥加密体系(Public Key Encryption System)中，加密过程和解密过程不是互逆的，所以加密密钥和解密密钥是两个不同的密钥。其中，由解密密钥可以求出加密密钥，但反过来，从加密密钥求出解密密钥则是非常困难的，在计算上是不可行的。这样，加密密钥可以被公开，供其他用户查阅，被称为公开密钥，以下简称公钥(Public Key，PK)；解密密钥则需要保密，仅为用户私人拥有，不能公开，被称为私有密钥，以下简称私钥(Secret Key，SK)。

当数据发送方 A 需要将消息 M 发送给接收方 B 时，可以用 B 的公钥 PK 将 M 加密，并将加密后的密文 C 传输给接收方 B，B 在收到密文 C 后，用自己的私钥 SK 进行解密，即可还原出原来的消息 M。

2. 公钥分发体系

公钥加密体系虽然具有对称加密体系所没有的优点，但相对于对称加密体系而言，它太复杂、运行速度太慢，这使得在实际系统中，公钥加密体系仍未直接应用于信息加密中。对称加密体系具有加密速度快、强度高等优点，因而在实际应用中需要使用信息加密时，一般仍采用对称加密体系。

对称加密体系虽然具有加密速度快、强度高等优点，但由于其加密和解密均采用一个共同的密钥，故共享密钥的管理和分发成了一个巨大的难题。在公钥密码体系未出现以前，人们使用基于可信的密钥分发管理中心 KDC 的密钥分发体系(Public Key Distribution System)。但由于KDC 保管着所有用户的密钥，当通信的双方需要共享秘密信息时，他们必须通过KDC 来实现通信密钥的共享，而这一过程非常繁琐，故网络的通信开销巨大。

公钥分发体系通过相应的分发协议解决了这一问题。它能够使得通信的双方经过比较简单的操作完成通信密钥的共享。

3. 数字签名体系

数字签名能够在网络通信中达到与现实世界中手写签名类似的效果。数字签名体系(Digital Signature System)采用签名密钥作为产生数字签名的唯一途径，利用验证公钥和相关算法来实现对消息来源的辨认和验证，并可以保证数据的完整性、真实性和不可抵赖性。

这样，数字签名技术能够满足上述的有关通信和信息传输安全需求中的真实性、完整性、不可否认性三种需求。它使得：

① 签名者发出签名的消息后，就不能再否认自己所签发的消息。

② 接收者能够确认或证实签名者的签名，核实消息的正确性，但不能否认。

③ 数字签名具有唯一性，除了签名者本人以外，任何人都不能伪造签名。

④ 第三方可以确认收发双方之间的消息传输，但不能伪造这一过程。当通信的双方关于签名的真伪发生争执时，可以由第三方来解决双方的争执。

对于公钥密码的三种体系而言，公钥加密体系由于性能问题在实际中很少被直接使用。为了满足通信和信息传输过程中的保密性需求，一般利用公钥分发体系使得通信的双方能够共享一个本次通信专用的临时密钥，然后利用这个临时密钥，结合对称密码体系实现保密通信。

所以说，密码技术是在公开信道上实现安全保密通信的核心技术之一。下一节将介绍公钥密码学的有关情况，并着重分析椭圆曲线公钥密码学的优势和特点。

1.4　公钥密码编码学

自1975年公钥密码编码的思想首次被提出以来，有许多的公钥密码体系被提出，但其中许多公钥密码体系是不安全的。另一些被认为是安全的公钥密码体系又有许多不实用的地方：要么密文扩展过于严重，要么密钥过大。对于那些既实用又安全的公钥密码体系而言，它们的安全基础都是某些复杂的含有陷门的数学难题，这些难题目前被公认为在计算上是不可行的。

根据公钥密码体系的安全基础来分类，现有的被公认为安全、实用、有效的公钥密码体系有三类。

① 基于大整数因式分解问题的IF 类公钥密码体系。这一类公钥密码体系基于大整数因式分解的困难性，其中，最著名的当属RSA 公钥密码体系。该算法是由麻省理工学院的Ron Rivest、Adi Shamir 和 Len Adleman 三位学者于 1978 年首次发表的(简称RSA)，是目前被广泛接受并实现的通用公开密码体系之一。

② 基于有限乘法群上离散对数问题的DLP 类公钥密码体系。DLP 类公钥密码体系基于有限乘法群上的离散对数问题的求解困难性，其中最著名的有 ElGamal 公钥密码体系和DSA 数字签名方案。

③ 基于代数曲线有限加法群上的离散对数问题的公钥密码体系。在这一类公钥密码体系中，最著名的就是椭圆曲线公钥密码体系 ECC，其数学基础是椭圆曲线有限加法群上的椭圆曲线离散对数问题的求解困难性。它是由华盛顿大学的Neal Koblitz 和IBM 公司的Victor Miller 于1986 年首次各自独立提出的。在同等安全条件下，椭圆曲线公钥密码体系具有许多独到的优势，因而成为当前的研究热点。

这里，从安全性和有效性两个方面对这些公钥密码体系进行分析。

1. 安全性分析

安全性分析目的在于分析破解某一密码体系所需要花费的代价，其中最重要的是计算安全性指标和密钥长度。

计算安全性指标的单位是MIPS 年，指的是每秒能执行100 万条整型计算指令的处理器不间断地运行一年所需要花费的计算工作量，该工作量大约相当于执行 3×10^{13}条指令。例如，一个 1GHz 的Pentium III 处理器可近似看成是一个200MIPS 的机器。密钥长度的单位是比特，这里假定密钥是完全随机生成的，因而具有最大

的熵值。

为了比较几种常用公钥密码体系的计算安全性，记

$$L_p(v,c) = O(e^{c(\ln p)^v(\ln \ln p)^{1-v}}) \tag{1.1}$$

为某一算法的计算复杂性表示。式中参数 c 是某个常数。

对于参数 $\ln p$，当 $v=0$ 时，$L_p(v,c)$ 与 $\ln p$ 是多项式关系；当 $v=1$ 时，是指数关系；当 $0<v<1$ 时，则它们是亚指数关系。

对于模数为 n 的 RSA 密码体系而言，目前最快的求解算法是数域筛法。用数域筛法分解大整数模数 n 的计算复杂性为 $L_p(1/3,c)$，即利用数域筛法能够在亚指数时间内攻破 RSA 密码体系。

同样，对于有限域 $GF(p)$ 上的 ElGamal 密码体系而言，目前最快的是指标积分算法(Index 算法)。用 Index 算法求解有限乘法群上离散对数问题 DLP 的平均计算时间复杂性为 $L_p(1/2,c)$，最好情况下可达到 $L_p(1/3,c)$。也就是说，存在着攻击 ElGamal 密码体系的亚指数时间算法。

而对于椭圆曲线密码体系而言，除了极少数特殊的椭圆曲线以外，求解有限域上的椭圆曲线离散对数问题 ECDLP 的最好算法的时间复杂度为 $L_p(1,c)$，即攻击椭圆曲线密码体系需要指数时间。

图 1-8 所示的为 RSA、ElGamal 和 ECC 三种密码体系的安全性比较，它反映了三种密码体系下计算安全性和密钥长度的关系。图中纵坐标表示的是密钥长度，而横坐标表示的是破解相应长度密钥所需要花费的计算工作量。

在目前的计算能力下，一般认为破译工作量超过 10^{12}MIPS 年时，该密钥长度是短期安全的；破译工作量超过 10^{15}MIPS 年时，该密钥长度是中期安全的；而为了保证长期安全性，密码体系的破译工作量不能少于 10^{20}MIPS 年。

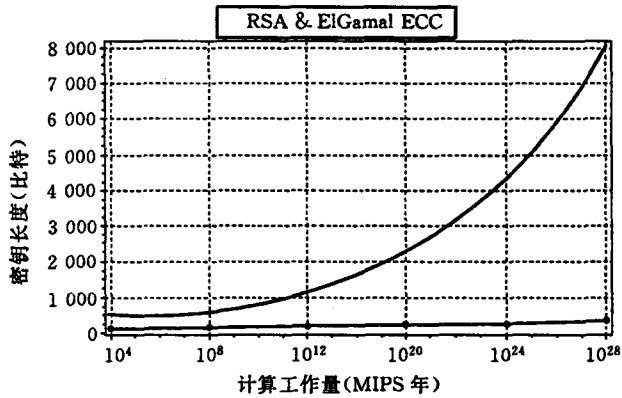

图 1-8 RSA、ElGamal 和 ECC 安全性的比较

从图 1-8 可以看出，当破译密码体系的计算工作量增加，也就是密码体系的安全等级提高时，RSA 和 ElGamal 密码体系中密钥长度的增长速度远远超过 ECC 密码体系的密钥增长速度。对于 RSA 和 ElGamal 密码体系而言，为了保证短期的安全性，所用的密钥长度不能少于1024 位；而160 位密钥长度对ECC 而言则已经足够了。为了保证长期的安全性，RSA 和ElGamal 密码体系需要至少2 048 位的密钥长度，而ECC 密码体系则只需要增长至210 位即可满足需求。

2. 有效性分析

一个公钥密码体系的有效性主要包括三个方面：计算开销、密钥长度和带宽需求。

设 N，n 分别表示具有相同计算复杂性的椭圆曲线系统和 ElGamal 系统相应的有限基域的规模，则 N 与 n 之间的关系为

$$N = 4.91n^{1/3}[\ln(n\ln 2)]^{2/3} \tag{1.2}$$

因此，对于相同规模的系统参数而言，椭圆曲线公钥密码体系

的每一比特密钥的强度要比其他两种密码体系大得多。也就是说，要得到同样强度的密码系统，椭圆曲线公钥密码体系的系统参数规模要小得多。

对于相同强度和安全等级的密码系统而言，由于椭圆曲线密码体系的密钥很短，所以其计算开销相应要小得多。由于椭圆曲线密码体系的安全性只与所选择的椭圆曲线有关，所以通过选择特殊的基域参数，可以进一步地提高椭圆曲线密码体系的实现速度。

当传输加密密文和签名消息，特别是长度较短的消息（如签名、会话密钥等）时，椭圆曲线密码体系的操作数较短，这使得它只需要很小的带宽需求。如果利用点压缩技术，还可以进一步地节省带宽资源。

较短的密钥和小规模的系统参数使得椭圆曲线公钥密码体系无论在实现上，还是在应用中都比其他类型的公钥密码体系具有更大的优越性，特别适用于像窄带拨号网络、网络智能卡、无线设备等计算能力、集成电路空间和带宽受限的场合，以及要求高速实现的情况。因而与其他密码体系相比，具有更强的竞争力，特别适用于网络经济和电子商务的信息安全防护。

第2章　椭圆曲线数学基础

椭圆曲线密码体系即基于椭圆曲线离散对数问题的各种公钥密码体系，是用有限域上的椭圆曲线有限群代替基于离散对数问题的密码体系中的有限循环群而得到的一类新型密码体系，是已有的各种密码编码方案在椭圆曲线上的实现。

自1985年数学家Neil Koblitz和Victor Miller各自独立地提出以椭圆曲线上的有理点构成的Abel群为背景结构、基于椭圆曲线离散对数问题的公钥密码体系以来，椭圆曲线密码体系逐步成为一个令人十分感兴趣的密码学分支。在椭圆曲线上实现各种已知的密码体系已是公钥密码学领域的一个重要课题，自1997年以来形成了一个研究热点。

椭圆曲线密码体系的理论基础是椭圆曲线数学理论和基于椭圆曲线离散对数问题。椭圆曲线理论起源于19世纪，在费尔马大定理的证明和因式分解等问题中起到了很重要的作用，多年来一直被认为是纯理论学科。

由于椭圆曲线领域所包含的内容非常庞杂，而现行的椭圆曲线密码体系只用到了椭圆曲线的少数几个特征，所以本章将着重介绍与椭圆曲线公钥密码体系相关的椭圆曲线数学理论基础以及椭圆群运算法则。

2.1 群

群(Groups)是抽象代数学的重要组成部分,是研究椭圆曲线密码体系的重要基础。限于篇幅,本节将简要介绍后继讨论中所需要的群的有关概念和性质。

定义 2.1 设有一个由任意元素 $a,b,c,\cdots$ 组成的非空集合 G,即 $G=\{g_i\}$。在 G 上有一个针对其中元素进行组合操作的二元运算规则"·",并同时满足下列四个条件,则 G 对于运算"·"称为群,并称二元运算"·"为群的运算。

① 封闭性 对于任意的 $g_i,g_j\in G$,有 $g_i\cdot g_j=g_k\in G$。即 G 中任意两个元素在所定义的群运算"·"下,依照次序合成的结果仍是 G 中的一个完全确定的元素。

② 结合律 对于任意的 $g_i,g_j,g_k\in G$,都有

$$(g_i\cdot g_j)\cdot g_k=g_i\cdot(g_j\cdot g_k)$$

③ 单位元 存在唯一的元素 $e\in G$,使得对于任意的 $g_i\in G$,满足

$$e\cdot g_i=g_i\cdot e=g_i$$

并称 e 为 G 的单位元素,简称单位元,有时也用 1 表示。

④ 逆元素 对于任意的 $g_i\in G$,都有相应的元素 $g_i^{-1}\in G$,使得

$$g_i\cdot g_i^{-1}=g_i^{-1}\cdot g_i=e$$

并称 g_i^{-1} 为 g_i 的逆元素,简称逆元。

上述四个条件是构成群的充分必要条件,通常称为群的公理。若仅满足条件①和条件②,则称为半群;满足条件①、②和条件③,称为幺半群。

定义 2.2 若群 G 对运算"·"还满足交换律,即对于任意的

$g_i, g_j \in G$，都有 $g_i \cdot g_j = g_j \cdot g_i$ 成立，则称群 G 为交换群或阿贝尔群（Abel Groups）。此时，通常用符号“+”来替代“·”，并称群运算“+”为“加法”，称 $a+b$ 为 a 与 b 的和，称单位元素 e 为零元素 O，称逆元素 a^{-1} 为元素 a 的负元素，并记作 $-a$。相应地称群运算“·”为“乘法”，称 $a \cdot b$ 为 a 与 b 的积，简写为 ab。例如，全体整数的集合在通常的加法运算下构成一个阿贝尔交换群。

定义2.3 群 G 中的元素个数称为群 G 的阶（Order），记作 $\#G$ 或 $|G|$。

定义2.4 若群 G 的阶为有限数，即群 G 中的元素个数 $|G|$ 有限，则称 G 为有限群；反之，称为无限群。

定义2.5 若无限群 G 中的群元素是可数无限的，则称 G 为离散群；反之，若无限群 G 中的群元素是不可数无限的，则称为连续群。

定义2.6 元素在数域 K 中的全体 n 级可逆矩阵对于矩阵乘法构成一个群，称这个群为 n 级一般线性群，记作 $GL_n(K)$；$GL_n(K)$ 中全体行列式为1的矩阵对于矩阵乘法也构成一个群，称为特殊线性群，记作 $SL_n(K)$。

定义2.7 对非空集合 S，可以证明 S 到自身的所有一一对应的映射组成的集合对于映射的复合运算构成一个群 G，称为对称群。其中，该群的单位元是恒等映射，并称群中的元素为 S 的一个置换。当 S 是 n 元有限集合时，称 G 为 n 元对称群，记作 S_n。

定义2.8 设 H 为群 G 的一个子集，若对于定义于群 G 上的二元运算规则“·”，H 满足由定义2.1所定义的群的公理，即 H 是一个群，则称 H 为群 G 的一个子群，记作 $H \subseteq G$。显然，$H=\{e\}$ 和 $H=G$ 都是群 G 的子群，称它们为群 G 的平凡子群；如果子群 H 不是群 G 的平凡子群，则称 H 为群 G 的真子群。

定义2.9 设 n 为任意正整数，对于任意的 $a \in G$，定义

$$a^n = \underbrace{a \cdot a \cdot \cdots \cdot a}_{n}, a^0 = e, \quad a^{-n} = (a^{-1})^n$$

并称a^n为a的n次幂，a^{-n}为逆元a^{-1}的n次幂，则对于任意整数m，有$(a^m)^n = a^{mn}$，$a^m a^n = a^{m+n}$。定义$\langle a\rangle$表示所有a^m的集合，则$\langle a\rangle$也构成一个有限群，并称$\langle a\rangle$中的元素为$\langle a\rangle$的生成元，称子群$\langle a\rangle$的阶为元素a的阶，记作ord(a)。特别地，若G中一个元素a，使得$\langle a\rangle = G$成立，则称G为循环群。

可以证明，循环群G都是阿贝尔群，循环群的子群仍是循环群。

定义2.10　设n为任意正整数，对于任意的$a \in G$，称满足$a^n = e$的最小正整数n为群元a的阶数。显然，对于有限群G而言，其每一群元的阶都是有限正整数。

定义2.11　设G, G'是两个群，f是从G到G'的一个映射。若对于任意的$a, b \in G$，都有

$$f(ab) = f(a)f(b)$$

那么，称f是G到G'的一个同态。

若映射f是一对一的映射，则称f为单同态；若映射f是满映射，则称f为满同态；若映射f是一一对应的映射，则称f为同构；当G到G'时，称同态f为自同态，同构f为自同构。

定义2.12　设G, G'是两个群，如果存在一个从群G到群G'的同构，则称为群G和群G'的同构，记作$G \cong G'$。

自19世纪中叶，由拉格朗日、阿贝尔、伽罗瓦等人引入群的概念以来，经过100多年的发展，群论已经成为现代代数学的重要分支，其内容非常丰富。下面，将根据本书的需要，介绍一些与椭圆曲线密码学有关的群的重要性质。

定理2.1(广义结合律)　对群中的任意n个元素$g_1, g_2, \cdots, g_n$，其积$g_1 \cdot g_2 \cdot \cdots \cdot g_n$唯一确定。

定理2.2　群的单位元e都是唯一的。

定理2.3　对于任意的$a,b,c\in G$，若$ab=ac$，则$b=c$；若$ab=cb$，则$a=c$。

定理2.4　群中每一元素的逆元是唯一的。

定理2.5　对阿贝尔交换群中的任意n个元素$g_1,g_2,\cdots,g_n$，及对$1,2,\cdots,n$的任意排列$i_1,i_2,\cdots,i_n$，有

$$g_{i_1}\cdot g_{i_2}\cdot\cdots\cdot g_{i_n}=g_1\cdot g_2\cdot\cdots\cdot g_n$$

定理2.6　对非空集合G及定义在其上的二元运算规则“$\cdot$”，若G是一个群，则方程

$$ax=b,\quad ya=b$$

在群G中有解；反之，若上述方程在非空集合G中有解，且二元运算规则“$\cdot$”满足结合律，则集合G是一个群。

定理2.7　加群Z的任一子群H都是循环群，并且有$H=\langle 0\rangle$或$H=\langle m\rangle=mZ$，其中m是H中的最小正整数；如果$H\neq 0$，则H是无限的。

定理2.8　每个无限循环群同构于加群Z，而每个阶为m的有限循环群则同构于加群Z/mZ。

定理2.9　设G是一个群，$a\in G$，

如果a是无限阶的，则

① 当且仅当$k=0$时，$a^k=e$；

② 元素$a^k(k\in Z)$两两不同。

如果a是有限阶的，其阶$m>0$，则

① m是使得$a^k=e$的最小正整数；

② 当且仅当$m\mid k$时，$a^k=e$；

③ 当且仅当$r\equiv k \bmod m$时，$a^r=a^k$；

④ 元素$a^k(k\in Z/mZ)$两两不同；

⑤ $\langle a\rangle=\{a,a^2,\cdots,a^{m-1},a^m=e\}$；

⑥ 对于任意整数 $1 \leqslant d \leqslant m$，有 $\mathrm{ord}(a^d) = \frac{m}{(m,d)}$。

2.2 环

和群一样，环(Ring)也是抽象代数学的重要组成部分之一，在椭圆曲线的研究中占有重要地位，是椭圆曲线密码系统的快速实现和解决计算有限域上的椭圆曲线群的阶等问题的核心之一。限于篇幅，本节将简要介绍后继讨论中所需要的环的有关概念和性质。

定义 2.13 设有非空集合 R，在 R 上定义了两个二元运算，一个是加法运算“+”，另一个是乘法运算“×”，则定义满足下列三个条件的代数结构 $<R,+,\times>$ 为环：

① R 上的元素关于加法运算 $(R,+)$ 构成阿贝尔群；

② R 关于乘法运算 $(R,\times)$ 构成半群；

③ 乘法运算对于加法运算满足分配律，即对于任意的 $a,b,c \in F$，有

$$(a+b)\times c = a\times c + b\times c$$

$$c\times(a+b) = c\times a + c\times b$$

若 $(R,\times)$ 是交换半群，则称 R 为交换环；若 $(R,\times)$ 有单位元，则称 R 为含单位元的环。全体整数、全体有理数、全体实数或全体复数在通常的加法和乘法运算下均构成环，分别称为整数环、有理数环、实数环和复数环，又可统称数环。

定义 2.14 设 a 是环 R 中的一个非零元，如果存在非零元 $b \in R$，使得 $ab=0$，则称 a 为环 R 的左零因子；相应地，若存在非零元 $c \in R$，使得 $ca=0$，则称 a 为环 R 的右零因子；如果 a 既是环 R 的

左零因子，又是环R的右零因子，则称a为环R的零因子。

定义2.15　若交换环R中有单位元，但没有零因子，则称R为整环。

定义2.16　设R为含单位元1_R的环，a是环R中的一个元，如果存在元$b\in R$，使得$ab=1_R$，则称a为b的左逆元；相应地，称b为a的右逆元。

定义2.17　设R是一个交换环，$a,b\in R$，且$b\neq 0$，若存在一个元素$c\in R$，使得$a=bc$，称b整除a，或者a被b整除，记作$b|a$；并称b为a的因子，a是b的倍元。若b,c均不是单位元，则称b为a的真因子。对元素$p\in R$，若p既不是单位元，也不是真因子，则称p为素元。

定义2.18　设R和R'是两个环，则称满足下列条件的映射f：$R\to R'$为同态环：

① 对于任意的$a,b\in R$，都有$f(a+b)=f(a)+f(b)$；

② 对于任意的$a,b\in R$，都有$f(ab)=f(a)f(b)$。

若f是一对一的映射，则称f为单同态；若f是满映射，则称f为满同态；若f是一一对应的映射，则称f为同构；如果存在一个从R到R'的映射，则f为环R和R'同构。

定义2.19　设R是一个环，若存在一个最小正整数n，使得$\forall\ a\in R$，都有$na=0$，则定义环R的特征为n；若不存在这样的正整数，则定义环R的特征为0。

定理2.10　对环R，有

① 对于任意的$a\in R$，有

$$0a = a0 = 0$$

② 对于任意的$a,b\in R$，有

$$(-a)b = a(-b) = -ab$$

③ 对于任意的$a,b\in R$，有

$$(-a)(-b) = ab$$

④ 对于任意的$n \in Z$，任意的$a, b \in R$，有

$$(na)b = na(b) = nab$$

⑤ 对于任意的$a_i, b_j \in R$，有

$$\left(\sum_{i=1}^{n} a_i\right)\left(\sum_{j=1}^{m} b_j\right) = \sum_{i=1}^{n}\sum_{j=1}^{m} a_i b_j$$

定理 2.11 设R为含单位元的环，n是正整数，且$a, b, a_1, \cdots, a_r \in R$。

① 若$ab=ba$，则

$$(a+b)^n = \sum_{k=0}^{n} \frac{n!}{k!(n-k)!} a^k b^{n-k}$$

② 若$a_i b_j = b_j a_i$ $(1 \leqslant i, j \leqslant r)$，则

$$(a_1 + \cdots + a_r)^n = \sum_{i_1 + \cdots + i_r = n} \frac{n!}{i_1! \cdots i_r!} a_1^{i_1} \cdots a_r^{i_r}$$

定理 2.12 设R为含单位元的交换环，若环R的特征是素数p，则对于任意的$a, b \in R$，有

$$(a+b)^p = a^p + b^p$$

定义 2.20 设多项式$f(x) = a_n x^n + \cdots + a_1 x + a_0$，$a_n \neq 0$，则称多项式$f(x)$的次数为$n$，记为$\deg f = n$。

定义 2.21 若由整环R上的全体多项式组成集合$R[X]$，设

$$\begin{cases} f(x) = a_n x^n + \cdots + a_1 x + a_0, & a_n \neq 0 \\ g(x) = b_n x^n + \cdots + b_1 x + b_0, & b_n \neq 0 \end{cases}$$

则定义$R[X]$上的加法为

$$(f+g)(x) = (a_n + b_n)x^n + \cdots + (a_1 + b_1)x + (a_0 + b_0)$$

显然，$R[X]$中的零元为0，$f(x)$中的负元为

$$(-f)(x) = (-a_n)x^n + \cdots + (-a_1)x + (-a_0)$$

类似地，可以定义$R[X]$上的乘法。设

$$\begin{cases} f(x) = a_n x^n + \cdots + a_1 x + a_0, & a_n \neq 0 \\ g(x) = b_m x^m + \cdots + b_1 x + b_0, & b_m \neq 0 \end{cases}$$

则定义$R[X]$上的乘法为

$$(f \cdot g)(x) = c_{n+m} x^{n+m} + \cdots + c_1 x + c_0$$

式中

$$c_k = \sum_{i+j=k} a_i b_j, 0 \leqslant k \leqslant n+m$$

即

$$\begin{cases} c_{n+m} = a_n b_m \\ c_{n+m-1} = a_n b_{m-1} + a_{n-1} b_m \\ \quad \vdots \\ c_1 = a_1 b_0 + a_0 b_1 \\ c_0 = a_0 b_0 \end{cases}$$

易知，$R[X]$中的单位元为1。

显然，由环的定义2.13、定义2.15可知，集合$R[X]$对于上述所定义的加法运算和乘法运算构成一个整环，称为多项式环。

多项式环有如下一些定义和性质。

定义2.22　设$f(x)$和$g(x)$是整环R上的任意两个多项式，其中$g(x) \neq 0$。若存在一个多项式$q(x) \in R$，使得等式$f(x) = g(x)q(x)$成立，则称$g(x)$整除$f(x)$，或者$f(x)$被$g(x)$整除，记作$g(x) \mid f(x)$，并称$g(x)$是$f(x)$的因式，$f(x)$是$g(x)$的倍式；否则，称$g(x)$不能整除$f(x)$，或者$f(x)$不能被$g(x)$整除，记作$g(x) \nmid f(x)$。

定义2.23　设$f(x)$是整环R上的非常数多项式，如果除了显然因式1和$f(x)$以外，$f(x)$没有其他因式，则称$f(x)$为不可约多项式；否则，称$f(x)$为合式。

定理2.13　设$f(x)$和$g(x)$是整环R上的两个多项式，且$\deg f$

$=n$,$\deg g=m$,$m>1$,则一定存在多项式$q(x)$和$r(x)$,使得

$$f(x)=g(x)q(x)+r(x),\deg r<\deg g$$

称$q(x)$为$f(x)$被$g(x)$除所得的不完全商,称$r(x)$为$f(x)$被$g(x)$除所得的余式。本定理是多项式欧几里得除法的基础。

类似于整数中的最大公约数和最小公倍数,这里给出多项式环$R[X]$中的相关定义。

定义 2.24 设$f(x)$和$g(x)\in R[x]$,则定义满足下列条件的多项式$d(x)\in R[x]$为$f(x)$和$g(x)$的最大公因式。

① $d(x)|f(x)$,且$d(x)|g(x)$;

② 若存在$h(x)\in R[x]$满足:$h(x)|f(x)$,且$h(x)|g(x)$,那么必有$h(x)|d(x)$。并记$f(x)$和$g(x)$的最大公因式为$(f(x),g(x))$。

定义 2.25 若$f(x)$和$g(x)$的最大公因式$(f(x),g(x))=1$,则称$f(x)$和$g(x)$是互质的,或者说是互素的。

定义 2.26 设$f(x)$和$g(x)\in R[x]$,则定义满足下列条件的多项式$D(x)\in R[x]$为$f(x)$和$g(x)$的最小公倍式:

① $f(x)|D(x)$,且$g(x)|D(x)$;

② 若存在$H(x)\in R[x]$满足$f(x)|H(x)$,且$g(x)|H(x)$,那么必有$D(x)|H(x)$,并记$f(x)$和$g(x)$的最小公倍式为$[f(x),g(x)]$。

2.3 域

与群和环一样,域(Field)也是现代代数学的重要组成部分,是解决椭圆曲线离散对数问题的理论基础之一,在椭圆曲线密码体系的研究中占有极其重要的地位。

定义 2.27 设F是一个至少包含两个元素的集合,在F上定

义了两个二元运算，一个是加法运算“+”，另一个是乘法运算“×”，则定义满足下列三个条件的代数结构$<F,+,\times>$为域。

① F 上的元素关于加法运算“+”构成阿贝尔群，设其零元素为O；

② $F\backslash\{O\}$关于乘法运算“×”也构成阿贝尔群。设其单位元为e，在不引起混乱的情况下可用1表示；

③ 乘法运算对于加法运算有分配律，即对于任意的$a,b,c\in F$，有

$$(a+b)\times c=a\times c+b\times c$$

$$c\times(a+b)=c\times a+c\times b$$

显然，域是环的一种类型，因而也是一种极其重要的代数结构。由于在域内，方程$a+x=b$和$ax=b(a\neq 0)$均有唯一解，故加法和乘法有逆运算，其逆运算分别为减法和除法，因而域是其内元素可做四则运算的代数结构。

定义2.28　若F是域E的一个子集合，它在域E的运算下也构成一个域，则称域F是域E的一个子域；而称域E是域F的一个扩域。

定义2.29　若域E是域F的扩域，则单位元$1_F=1_E$，且域E可作为域F上的线性空间，此时，可用$[E:F]$表示域F在域E上线性空间的维数。如果$[E:F]$是有限的，则称域E为域F的有限扩张；否则，若$[E:F]$是无限的，则称域E为域F的无限扩张。

定理2.14　设域E是域F的扩域，域F是域K的扩域，则

$$[E:F]=[E:F][F:K]$$

如果$\{\alpha_i\}_{i\in I}$是域F在域K上的基底，$\{\beta_j\}_{j\in J}$是域E在域F上的基底，则$\{\alpha_i\beta_j\}_{i\in I,j\in J}$是域$E$在域$K$上的基底。

首先，对于任意的$c\in E$，由于$\{\beta_j\}_{j\in J}$是域E在域F上的基底，故存在$b_j\in F,j\in J$，使得

$$c = \sum_{j\in J} b_j\beta_j$$

又因为$\{\alpha_i\}_{i\in I}$是域F在域K上的基底，所以存在$a_{ij}\in K, i\in I$，使得

$$b_j = \sum_{i\in I} a_{ij}\alpha_i$$

从而

$$c = \sum_{i\in I, j\in J} a_{ij}\alpha_i\beta_j$$

即$\{\alpha_i\beta_j\}_{i\in I, j\in J}$是域$E$在域$K$上的生成元。

另一方面，若存在$a_{ij}\in K, i\in I$，使得

$$\sum_{i\in I, j\in J} a_{ij}\alpha_i\beta_j = 0 \quad 或 \quad \sum_{j\in J}\Big(\sum_{i\in I} a_{ij}\alpha_i\Big)\beta_j = 0$$

成立，则由于$\{\beta_j\}_{j\in J}$是域E在域F上的基底，所以

$$\sum_{i\in I} a_{ij}\alpha_i = 0 \qquad (j \in J)$$

又因为$\{\alpha_i\}_{i\in I}$是域F在域K上的基底，所以

$$a_{ij} = 0 \qquad (i \in I, j \in J)$$

即$\{\alpha_i\beta_j\}_{i\in I, j\in J}$在域$K$上是线性无关的。从而，有

$$[E:F] = [E:F]\,[F:K]$$

由本定理可以得出如下推论。

推论2.1 域E是域K的有限扩域的充分必要条件是：域E是域F的有限扩域，且域F是域K的扩域。

定义2.30 设F是一个域，$X\subset F$，则包含X的所有子域的交集仍是包含X的子域，称为X生成的子域。若域F是域K的扩域，且$X\subset F$，则称由$K\cup F$生成的子域为X在域K上生成的子域，记作$K(X)$。对应地，包含X的所有子环的交集仍是包含X的子环，称为X生成的子环；并称由$K\cup F$生成的子环为X在K上生成的子环，记作$K[X]$。$K[X]$是一个整环。

定义2.31　如果$X=\{u_1,\cdots,u_n\}$，则将域F的子域$K(X)$记作$K(u_1,\cdots,u_n)$，并称域$K(u_1,\cdots,u_n)$为K的有限扩张；若$X=\{u\}$，则称$K(u)$为K的单扩张。

定理2.15　设域F是域K的扩域，$u,u_1,\cdots,u_n\in F$，$X\subset F$，则

① 子环$K(u)$由形如$f(u)$的元素组成，其中f是系数在K上的多项式，即$f\in K[x]$；

② 子环$K(u_1,\cdots,u_n)$由形如$f(u_1,\cdots,u_n)$的元素组成，其中f是系数在K上的n元多项式，即$f\in K[x_1,\cdots,x_n]$；

③ 子环$K[x]$由形如$f(u_1,\cdots,u_n)$的元素组成，其中，

$$n\in N,u_1,\cdots,u_n\in X,f\in K[x_1,\cdots,x_n]$$

④ 子域$K(u)$由形如$\dfrac{f(u)}{g(u)}$的元素组成，其中$f,g\in K[x]$，$g(u)\neq 0$；

⑤ 子域$K(u_1,\cdots,u_n)$由形如$\dfrac{f(u_1,\cdots,u_n)}{g(u_1,\cdots,u_n)}$的元素组成，其中，

$$f,g\in K[x_1,\cdots,x_n],\quad g(u_1,\cdots,u_n)\neq 0$$

⑥ 子域$K(X)$由形如$\dfrac{f(u_1,\cdots,u_n)}{g(u_1,\cdots,u_n)}$的元素组成，其中，

$$n\in N,f,g\in K[x_1,\cdots,x_n],u_1,\cdots,u_n\in X$$
$$g(u_1,\cdots,u_n)\neq 0$$

⑦ 对于每一个$v\in K[X]$，存在一个有限子集$X'\subset X$，使得$v\in K[X']$。

定义2.32　设域F是域K的一个扩域，如果存在一个非零多项式$f\in K[X]$，使得$f(u)=0$，则称域F中的元素u为域K上的代数数；反之，如果不存在任何非零多项式$f\in K[X]$，使得$f(u)=0$，则称域F中的元素u为域K上的超越数；如果域F中的每个元素都是域K上的代数数，则称域F为域K上的代数扩张；如果域F中至少有一个元素是超越数，则称域F为域K上的超越扩张。

定理 2.16 设域 F 是域 K 的扩域，$u\in F$ 是 K 上的超越数，则存在一个在域 K 上为恒等映射的域同构 $K(u)\cong K(x)$。

定理 2.17 设域 F 是域 K 的扩域，$u\in F$ 是 K 上的代数数，则存在唯一的在域 K 上的首一不可约多项式 $f(x)$，使得 $f(u)=0$。此时的首一不可约多项式 $f(x)$ 称为 u 的不可约多项式，也称为极小多项式，u 在 K 上的次数为 $\deg f$，u 的极小多项式的其他根称为 u 的共轭根。

定义 2.33 设 F 是一个域，$f\in F[x]$ 是 F 上的多项式，且 $\deg f\geqslant 1$；设域 E 是域 F 的一个扩域，如果 f 在 $E[x]$ 中可分解，且 $E=F(u_1,\cdots,u_n)$，其中，$u_1,\cdots,u_n$ 是 f 在 E 中的根，则称此时的域 E 为多项式 f 在 F 上的分裂域。

定义 2.34 设域 E 是域 F 的扩域，若 E 中的元都是 F 的代数元，则称域 E 是域 F 的代数扩域。而没有真代数扩域的域称为代数闭域，即域中的不可约多项式都是一次多项式。对于任意域 F，若存在代数闭域 $\overline{F}$，则称 $\overline{F}$ 是 F 的代数闭包。

定义 2.35 对于任意一个抽象的域 F，考虑由单位元 e 生成的加法群，即 $\{0,\pm e,\pm 2e,\pm 3e,\cdots\}$，则它有下面两种可能。

① 对于任意的正整数 n，有 $ne\neq 0$，也就是说，域 F 是一个无限循环群。在这一情形下，域 F 包含所有的商 ne/me，其中 n、m 为整数，且 $m\neq 0$。显然，这种元素的全体构成了域 F 的一个子域，而该子域可以与有理数域等同起来。这时，称域 F 的特征为零，记作 $\mathrm{Char}(F)=0$。

② 存在一个最小的正整数 p，使得 $pe=0$。容易证明，满足这一条件的 p 一定是素数。而此时的域 $\{0,e,\cdots,(p-1)e\}$ 已经构成了域 F 的一个子域。该子域可以与整数模 p 的域 F_p 等同起来。这时，称域 F 的特征为 p，记作 $\mathrm{Char}(F)=p$。

2.4 有 限 域

只含有有限个元素的域叫做有限域(Finite Field)。由于它首先由E. 伽罗瓦所发现,因而又称为伽罗瓦域(Galois Field)。在同构意义下,对于任一素数p和正整数n,存在且仅存在一个含p^n个元素的有限域,记作$GF(p^n)$。有限域$GF(p^n)$的特征为p,其阶为域中元素的个数,即p^n。另一方面,对$q>1$的整数而言,q阶有限域$GF(q)$存在的充要条件是q为某一素数的整次幂(以下简称素数幂)。

为了方便讨论有限域$GF(p^n)$,首先需要了解有关整数及其模运算的基本概念和性质。

定义2.36 设a,b是任意两个整数,其中$b\neq 0$。若存在一个整数q,使得等式$a=bq$成立,这时,就称b整除a或者a被b整除,记作$b|a$,并把b称为a的因数,把a称为b的倍数。这时,q也是a的因数,常常将q写成ab^{-1}。否则,就称b不能整除a或者a不能被b整除。

定义2.37 给定一个正整数m,若两整数a,b满足$m|a-b$(即$a-b$被m整除),则称整数a,b对模m同余,记作$a\equiv b \bmod m$。

同余是整数间的一个等级关系。利用同余的定义,可以得到如式(2.1)和式(2.2)所示的模运算法则。

① 若$a_0\equiv b_0 \bmod m$,$a_1\equiv b_1 \bmod m$,则下列公式成立:

$$\begin{aligned} a_0+a_1 &\equiv (b_0+b_1) \bmod m \\ a_0-a_1 &\equiv (b_0-b_1) \bmod m \\ a_0a_1 &\equiv b_0b_1 \bmod m \end{aligned} \tag{2.1}$$

② 若$ac\equiv bc \pmod m$,则有

$$a \equiv b \bmod \frac{m}{(c,m)} \tag{2.2}$$

式中，(c,m)为整数c与m的最大公约数。

③ 若$a \equiv bc \bmod m$，则称c是a与b的模商，即$c \equiv ab^{-1} (\bmod m)$。特别地，若$ab \equiv 1 \bmod m$，则称$b$是$a$的逆。

根据模运算法则可直接在整数环Z模m得到的集合$Z_m = \{0, 1, \cdots, m-1\}$上实施四则运算。

上述的整数同余和模运算定义可以类推到一元代数多项式上，对于多项式环$R[x]$，设有多项式$a(t)$对$m(t)$求模，可用定理2.13所描述的多项式欧几里得多项式除法求$a(t)$除以$m(t)$的余式$r(t)$，其中$r(t)$的次数必须低于$m(t)$。这一操作可用式(2.3)来表示。

$$r(t) = a(t) \bmod m(t) \tag{2.3}$$

在式(2.3)中，若余式$r(t)$为0，则称$a(t)$整除$m(t)$。

定义2.38 对于多项式环$R[x]$，设有两个多项式$a(t)$和$b(t) \in R[t]$，若它们对给定的首一多项式$m(t)$有同样的求模余式，即$m(t) | (a(t) - b(t))$，则称多项式$a(t)$和$b(t)$对模多项式$m(t)$同余。这一关系可用式(2.4)来表示。

$$a(t) \equiv b(t) \bmod m(t) \tag{2.4}$$

显然，$R[x]$中的任意多项式$a(x)$都与其被$m(x)$除的余式$r(x)$模$m(x)$同余，并称该余式为$f(x)$模$m(x)$的最小余式，记为$f(x) \bmod m(x)$。

同样，可以得到类似于式(2.1)和式(2.2)所示的多项式模运算法则，这是下面将要介绍的有限域$GF(p^n)$的多项式等价类表示法的基础。

一般地，有限域$GF(p^n)$能被描述为系数属于$GF(p)$的多项式

等价类，也可在同构意义上通过任一n次不可约多项式产生。

例如，对域$GF(2^3)$而言，其模多项式可以是x^3+x^2+1，x^3+x+1，也可以是其他的三次不可约多项式。若选取x^3+x+1作为模多项式，则可以用低于三次的多项式来表示$GF(2^3)$中的元素0，x^0，x^1，…。具体操作如下。

$$x^3\equiv -x-1\equiv x+1$$

$$x^4\equiv x(x^3)\equiv x(x+1)\equiv x^2+x$$

$$x^5\equiv x(x^2+x)\equiv x^3+x^2\equiv x^2-x-1\equiv x^2+x+1$$

$$x^6\equiv x(x^2+x+1)\equiv x^3+x^2+x\equiv x^2-x\equiv x^2+1$$

$$x^7\equiv x(x^2+1)\equiv x^3+x\equiv -1\equiv 1\equiv x_0$$

表2.1描述了有限域$GF(2^3)$上元素的几种不同表示。表中各列分别表示元素的乘幂、多项式、由多项式系数所组成的三元向量组以及对应于该向量的常规表达。其中，由0和1组成的n元向量组有时也称为位字符串。

表2.1 有限域$GF(2^3)$上元素

乘幂	多项式	系数向量组	十进制数
0	0	(000)	0
x^0	1	(001)	1
x^1	x	(010)	2
x^2	x^2	(100)	4
x^3	$x+1$	(011)	3
x^4	x^2+x	(110)	6
x^5	x^2+x+1	(111)	7
x^6	x^2+1	(101)	5

显然，表2.1中的多项式集合对模x^3+x+1在加法和乘法运算上都是封闭的。这时$GF(2^3)$称为$GF(2)$的扩展域，而$GF(2)$则是$GF(2^3)$的基域。

若一个不可约多项式能够按上述方式生成域中的所有元素，则称它为简单不可约多项式。对于任意的素数p或素数幂q和正整数n，存在一个定义在$GF(q)$上的n次简单不可约多项式。

虽然对每一个素数的乘幂p^n来说，都存在一个同构意义上的确定的有限域$GF(p^n)$，但在本书的研究中，除非特别说明，将只讨论与椭圆曲线密码学相关的下列三种类型的有限域$GF(q)$。

① 当q是一个素数p时，域$GF(p)$称为素数有限域。典型的素数有限域就是由整数环Z模素数p而得到的p个元素$0,1,\cdots,p-1$组成的集合$Z_p=\{0,1,\cdots,p-1\}$，按模p运算法则所组成的有限域一般记作$GF(p)$或F_p。

② 对某些m，当$q=2^m$时，域$GF(2^m)$称为二次有限域。与素数有限域不同，二元有限域有许多常用的表示方法，在本书中一般用位字符串来表示。

③ 对某些奇素数p和m，当$q=p^m$时，域$GF(p^m)$称为奇特征扩张有限域。$GF(p^m)$通常用多项式等价类来表示，即模简单不可约域多项式的余式的集合，它类似于表2.1中的向量组。

这种有限域类似于上面两种有限域的结合，Aoki、Bailey等人指出：在用硬件实现ECC，且p的比特位数与处理器的字长接近时，利用$GF(p^m)$上的椭圆曲线密码是比较合适的。但在后面也将介绍到，这种有限域上的椭圆曲线密码体系更易受到攻击。

2.5 椭圆曲线

由代数几何学可知，椭圆曲线(Elliptic Curve，EC)是亏格

(Genus)为1的代数曲线。根据密码体系的研究需要,本节所要讨论的是光滑的椭圆曲线。这一曲线在解析平面中表现为一条非奇异(Nonsingularity)的三次平面曲线。由亏格公式可知:光滑的椭圆曲线一定是三次曲线;反之,光滑的三次曲线也一定是椭圆曲线。由Riemman-Roch 定理和亏格公式可知,任何一条椭圆曲线都可以用一个三次方程来表示,这个三次方程一般称为Weierstrass方程。

因此,椭圆曲线并非椭圆,这样命名的原因是因为对椭圆曲线的研究来源于椭圆周长计算问题,以及式(2.5)所描述的椭圆积分等问题。这里$E(x)$是x的三次或四次多项式。

$$\int \frac{\mathrm{d}x}{\sqrt{E(x)}} \tag{2.5}$$

由于这种类型的椭圆积分是不能用初等函数来表达的,为此人们引进了所谓的椭圆曲线函数。所谓椭圆曲线,就是由式(2.6)所描述的Weierstrass 方程所确定的平面曲线。椭圆曲线通常用E表示。

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{2.6}$$

定义2.39 设有给定域F,$\overline{F}$是F的一个代数闭域,对于三次齐次方程

$$\begin{aligned} &Y^2Z + a_1XYZ + a_3YZ^2 \\ &= X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3 \quad (a_i \in F) \end{aligned} \tag{2.7}$$

假设其满足Jacobi 条件,即若令

$$\begin{aligned} G(X,Y,Z) =& Y^2Z + a_1XYZ + a_3YZ^2 \\ &- (X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3) \end{aligned}$$

方程组

$$\begin{cases} \partial F/\partial X = 0 \\ \partial F/\partial Y = 0 \\ \partial F/\partial Z = 0 \\ G(X,Y,Z) = 0 \end{cases} \tag{2.8}$$

在域$\overline{F}$上无解，则定义于域F上的椭圆曲线E即方程式(2.7)在射影平面$P^2(\overline{F})$上的解集合为

$E=\{(X,Y,Z)\in P^2(\overline{F})\mid G(X,Y,Z)=0$，且$X,Y,Z$不全为零$\}$

式中，射影平面$P^2(\overline{F})$是指$F^3\backslash\{(0,0,0)\}$中按等价关系“～”所导出的全体等价类。

在定义2.39中，当方程组(2.8)无解时，称E是一条非奇异的(Nonsingularity)或者光滑的(Smooth)椭圆曲线，反之，则称其为奇异的椭圆曲线。

由方程式(2.7)可知，当$Z=0$时，椭圆曲线E上对应的有理点为$(0,1,0)$；当$Z\neq 0$时，对方程式(2.7)两端同时除以Z^3，可得

$$\left(\frac{Y}{Z}\right)^2 + a_1\left(\frac{XY}{Z^2}\right) + a_3\left(\frac{Y}{Z}\right) = \left(\frac{X}{Z}\right)^3 + a_2\left(\frac{X}{Z}\right)^2 + a_4\left(\frac{X}{Z}\right) + a_6 \tag{2.9}$$

若令$x=\frac{X}{Z}$，$y=\frac{Y}{Z}$，则方程式(2.9)将化为

$$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6 \quad (a_i\in F) \tag{2.10}$$

显然式(2.10)即方程式(2.6)，(x,y)和(X,Y,Z)是一一对应的，而且对于任取的一非零常数λ，(X,Y,Z)和$(\lambda X,\lambda Y,\lambda Z)$表示同一个点。

对比方程式(2.6)和方程式(2.7)，二者的解集只相差一个特殊的有理点$(0,1,0)$，可以把它看作是点$(0,1,\varepsilon)(\varepsilon\to 0)$，相应地可以理解为沿$y$轴趋向无穷远，所以特殊点$(0,1,0)$又被称为无穷远点(Point at Infinity)或零点(Zero Point)，记为O。

对于椭圆曲线E，可用方程式(2.6)外加一个特殊点O来表示。这时，称方程式(2.6)为椭圆曲线E的仿射方程，而称满足方程式(2.6)的数偶(x,y)为F域上椭圆曲线E上的点。其中，F域是代数闭域，它既可以是有理数域R，也可以是复数域C，还可以是有限域$GF(p^n)$。与之对应，方程式(2.7)又被称为椭圆曲线E的射影方程。

2.6　椭圆曲线的分类

设椭圆曲线E的仿射方程如方程式(2.6)，若令

$$\begin{cases} b_2 = a_1^2 + 4a_2 \\ b_4 = a_1a_3 + 2a_4 \\ b_6 = a_3^2 + 4a_6 \\ b_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \end{cases} \tag{2.11}$$

$$\begin{cases} c_4 = b_2^2 - 24b_4 \\ c_6 = b_2^3 + 36b_2b_4 - 216b_6 \end{cases} \tag{2.11}'$$

以及

$$\begin{cases} \Delta(E) = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6 \\ j(E) = c_4^3/\Delta(E) \end{cases} \tag{2.12}$$

显然，当且仅当方程式(2.12)中的$\Delta(E)\neq 0$时，椭圆曲线方程式(2.6)是光滑的，或者称为非奇异的。在本书的后继讨论中，均约定$\Delta(E)\neq 0$。

定义2.40　设有椭圆曲线E，其仿射方程如方程式(2.6)所示，则称方程式(2.12)所定义的$\Delta(E)$为椭圆曲线E的判别式，而$j(E)$为椭圆曲线E的j不变量，它是椭圆曲线E的一个同构不变

量。若定义在域 F 上的两条椭圆曲线同构，则它们的 j 不变量相同；反之，两条具有相同 j 不变量的椭圆曲线一定在域 $\overline{F}$ 上同构，其中 $\overline{F}$ 是 F 的代数闭域。

定义 2.41 设有两条椭圆曲线 E 和 E'，其中 E 的仿射方程如方程式(2.6)所示，而 E' 的仿射方程如下：

$$y'^2 + a'_1x'y' + a'_3y' = x'^3 + a'_2x'^2 + a'_4x' + a'_6 \quad (a'_i \in F)$$

若存在 $r,s,t,u \in F, u \neq 0$，使得方程式(2.6)的解 (x,y) 和上述方程的解 (x',y') 之间存在对应关系

$$\begin{cases} x = u^2x' + r \\ y = u^3y' + su^2x' + t \end{cases} \tag{2.13}$$

则称椭圆曲线 E 和 E' 在域 F 上是同构的。

按照定义 2.41，对于任意给定的一条椭圆曲线 E，总可以选取一组适当的变量代换式(2.13)，使曲线 E 在仿射坐标下的 Weierstrass 方程式(2.6)具有更加简单的形式。下面将根据域 F 的特征 Char(F) 和 j 不变量，给出不同情况下的约简形式。

① Char(F)$\neq$2,3。这时，令

$$\begin{cases} x = x' - \dfrac{1}{12}b_2 \\ y = y' - \dfrac{a_1}{2}\left(x' - \dfrac{1}{12}b_2\right) - \dfrac{1}{3}a_3 \end{cases}$$

则一般形式的 Weierstrass 方程式(2.6)总可以化为如下形式：

$$y^2 = x^3 + ax + b \quad (a,b \in F) \tag{2.14}$$

这里，为简单起见，用 x,y 替代了 x',y'（以下类似）。例如，当 $a=-3, b=3$ 时，方程式(2.14)的图像如图 2-1 所示。

② Char(F)=2。这时需要分两种情况对方程式(2.6)进行化

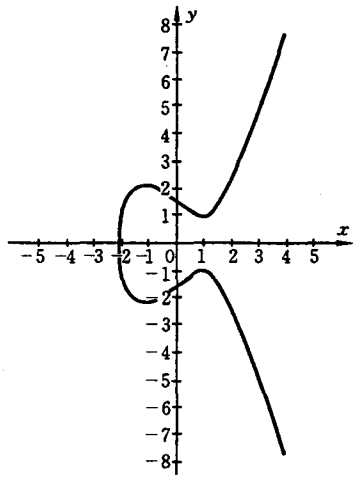

图2-1　椭圆曲线 $y^2=x^3-3x+3$ 的图像

简。

(a) $j(E)\neq 0$，即 $a_1\neq 0$。此时，令

$$\begin{cases} x = a_1^2 x' + \dfrac{a_3}{a_1} \\ y = a_1^3 y' + \dfrac{a_1 a_4 + a_3^2}{a_1^3} \end{cases}$$

则方程式(2.6)可化为

$$y^2 + xy = x^3 + a_2 x^2 + a_6 \qquad (2.15)$$

式中，$a_2, a_6 \in F$。对于方程式(2.15)，有 $\Delta(E)=a_6$，$j(E)=1/a_6$。

例如，当 $a_2=-3$，$a_6=2$ 时，椭圆曲线方程式(2.15)为

$$y^2 + xy = x^3 - 3x^2 + 2。$$

其图像如图2-2所示。

(b) $j(E)=0$，即 $a_1=0$。这时，一般令

$$\begin{cases} x = x' + a_2 \\ y = y' \end{cases}$$

则方程式(2.6)可化为

$$y^2 + a_3 y = x^3 + a_4 x + a_6 \tag{2.16}$$

式中，$a_4, a_6 \in F$。对于方程式(2.16)，有 $\Delta(E) = a_3^4$，$j(E) = 0$。

例如，当 $a_3 = 2$，$a_4 = -3$，$a_6 = 2$ 时，方程式(2.16)的图像如图2-3所示。

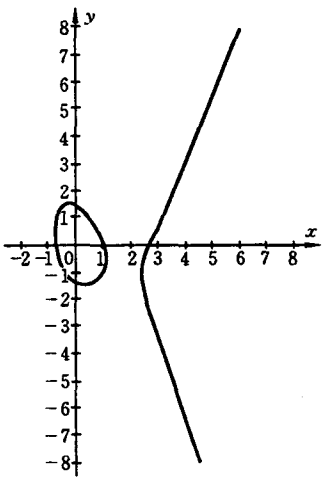

图 2-2　$y^2+xy=x^3-3x^2+2$ 的图像

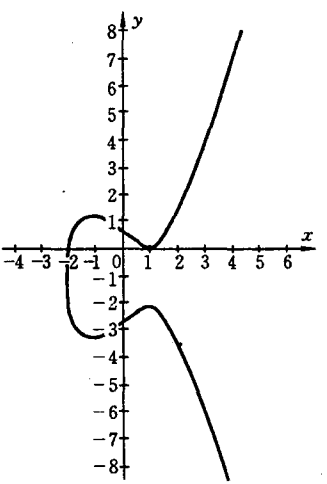

图 2-3　$y^2+2y=x^3-3x+2$ 的图像

③ Char(F)＝3。这时，同样也需要分两种情况对方程式(2.6)进行化简。

(a) $j(E) \neq 0$，即 $a_2 \neq 0$。可令

$$\begin{cases} x = x' + \dfrac{a_4}{a_2} \\ y = y' - \dfrac{1}{2}a_1x' - \dfrac{a_1a_4 + a_2a_3}{2a_2} \end{cases}$$

则方程式(2.6)可化为

$$y^2 = x^3 + a_2x^2 + a_6 \tag{2.17}$$

式中，$a_2, a_6 \in F$。对于方程式(2.17)，有$\Delta(E) = -a_2^3a_6$，$j(E) = -\dfrac{a_2^3}{a_6}$。

例如，当$a_2 = -3$，$a_6 = 3$时，曲线方程$y^2 = x^3 - 3x^2 + 3$的图像如图2-4所示。

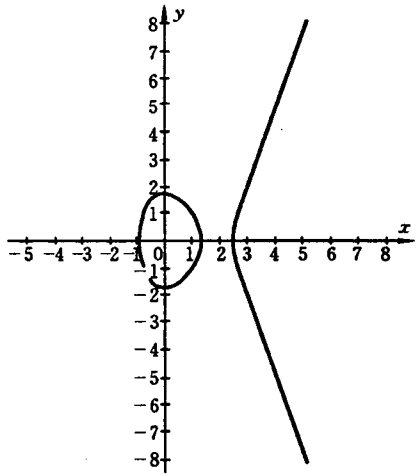

图2-4　椭圆曲线$y^2 = x^3 - 3x^2 + 3$的图像

(b) $j(E) = 0$，即$a_2 = 0$。令

$$\begin{cases} x = x' \\ y = y' - \dfrac{1}{2}(a_1x' + a_3) \end{cases}$$

则方程式(2.6)可化为

$$y^2 = x^3 + a_4x + a_6 \tag{2.18}$$

式中，$a_4, a_6 \in F$。对于方程式(2.18)，有 $\Delta(E) = -a_4^3$，$j(E) = 0$。

在本书以后的讨论中，当需要设定一条椭圆曲线的方程时，若对域的情况已有所约定，则将根据域的Char(F)和曲线的j不变量等情况，选取方程式(2.14)～式(2.18)中某一个作为该曲线的Weierstrass 方程进行研究。

2.7 椭圆曲线上点的群运算法则

由代数几何理论可知，在任一椭圆曲线E的点之间，存在一个可以由该曲线的除子类群和函数域构造的自然群运算法则。这时，由椭圆曲线E可构造一个椭圆曲线群，椭圆曲线E上的点对应于椭圆曲线群中的元素。即对椭圆曲线E上的任意两点$P, Q \in E$，由Riemman-Roch 定理可知，一定存在另外一点$R \in E$，使得$(P)+(Q) \sim (R)+(O)$。这里(P)、(Q)、$(P)+(Q)$等为椭圆曲线E的除子，而符号"～"则表示除子之间的一种等价关系。这时，称点R为点P和Q的和，记为$R = P+Q$。椭圆曲线E构成一个Abel 加法群，O为其单位元。简单地说，椭圆曲线群的这一运算法则就是：若椭圆曲线上的三个点处于一条直线上，那么它们的和为O。

由此，可以得出下列有关椭圆曲线群的具体群运算规则。

① 对于椭圆曲线上的任何一点P，有$P+O=P$；

② 在仿射坐标系上的一条平行于y轴的直线与椭圆曲线相交于三个点：两个有相同x坐标的点(设为P, Q)和无穷远点O。因此，$P+Q+O=O$。于是，有$-P=Q$。即作为群元素P的逆元Q，在仿射坐标系中与群元P的x坐标相同。这一关系如图2-5所示。

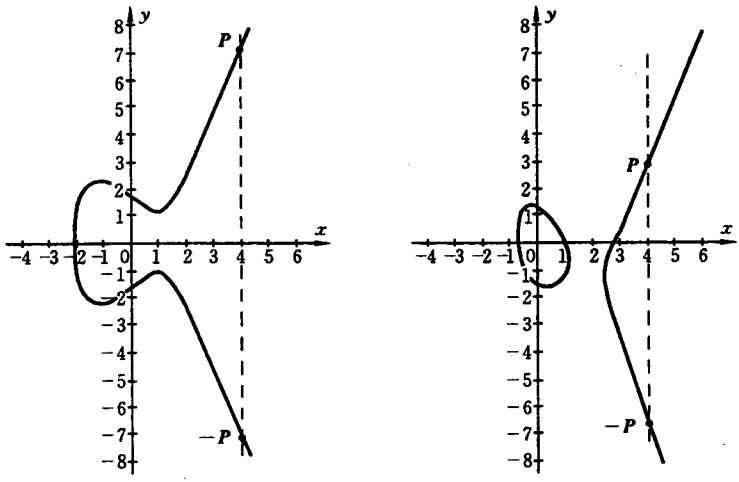

图2-5　椭圆曲线上的点P与其逆元$-P$

③ 椭圆曲线上的两个不同的点P与Q和$P+Q$是满足下列条件的椭圆曲线上的点R：P、Q和$-R$位于一条直线上。图2-6描述了这一操作。

由于该操作的操作对象和运算结果均为椭圆曲线上的点，因此，该操作有时又简称为点加运算。显然，点加运算满足交换律和结合律。

④ 在上述运算规则中，若$P=Q$，则称为倍点运算。倍点$R=2P$的逆元$-R$在过点P的椭圆曲线E的切线上。图2-7演示了这一操作规则。

由于密码学中主要讨论的是仿射坐标系下的椭圆曲线E，因此下面将根据上述群运算规则，给出仿射坐标系下的计算公式。

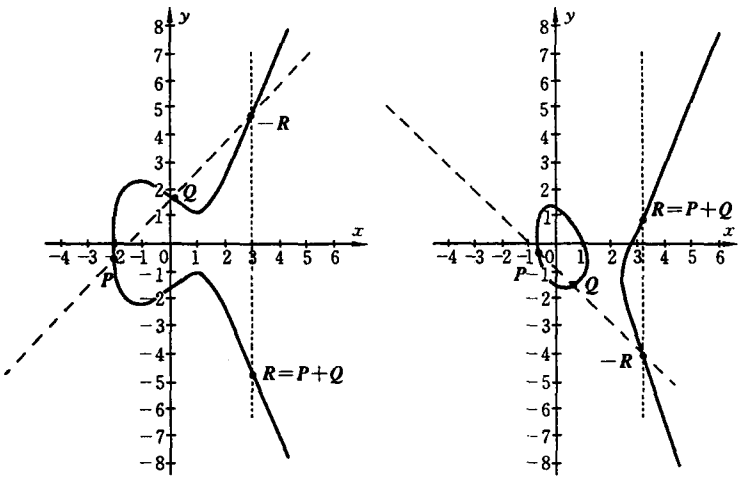

图 2-6　椭圆曲线上的点 P 与 Q 的加法运算图示

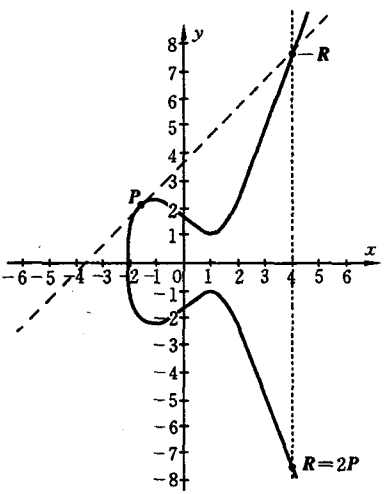

图 2-7　椭圆曲线上的点 P 的倍点运算图示

1. 逆元

设$P=(x,y)\in E$，根据Weierstrass 方程式(2.6)，由根与系数的关系易知

$$-P=(x,-y-a_1x-a_3) \tag{2.19}$$

2. 和

现设椭圆曲线E上任意两个非零点$P=(x_1,y_1)$，$Q=(x_2,y_2)$，且有$P\neq -Q$，则设$R=(x_3,y_3)$是P与Q的和，L是PQ连线，则$-R$在直线L上。

设过P，Q的直线L的方程为

$$y=\lambda x+v \tag{2.20}$$

式中，

$$\lambda=\begin{cases}\dfrac{y_2-y_1}{x_2-x_1} & \text{当 } P\neq Q \text{ 时}\\ \dfrac{3x_1^2+2a_2x_1+a_4-a_1y_1}{2y_1+a_1x_1+a_3} & \text{当 } P=Q \text{ 时}\end{cases} \tag{2.21}$$

$$v=\begin{cases}\dfrac{y_1x_2-y_2x_1}{x_2-x_1} & \text{当 } P\neq Q \text{ 时}\\ \dfrac{-x_1^3+a_4x_1+2a_6-a_3y_1}{2y_1+a_1x_1+a_3} & \text{当 } P=Q \text{ 时}\end{cases} \tag{2.22}$$

将式(2.20)代入一般形式的Weierstrass 方程式(2.6)，可得

$$(\lambda x+v)^2+a_1x(\lambda x+v)+a_3(\lambda x+v)=x^3+a_2x^2+a_4x+a_6$$

整理后可得

$$x^3-(\lambda^2+a_1\lambda-a_2)x^2-(2\lambda v+a_1v+a_3\lambda-a_4)x-(v^2+a_3v-a_6)=0 \tag{2.23}$$

由于P，Q和$-R$都在直线L上，所以，x_1、x_2和x_3都是方程式

(2.23)的根，由代数方程的根与系数的关系可知

$$x_1 + x_2 + x_3 = \lambda^2 + a_1\lambda - a_2$$

所以，结合式(2.19)、(2.20)，可知点 R 的坐标(x_3, y_3)如下：

$$\begin{cases} x_3 = \lambda^2 + a_1\lambda - a_2 - x_1 - x_2 \\ y_3 = -(\lambda + a_1)x_3 - v - a_3 \end{cases} \tag{2.24}$$

式中，λ、v 分别由式(2.21)和式(2.22)确定。

椭圆曲线上的点加运算的结果一定是椭圆曲线上的一个点。反之，若有一条曲线上的点加运算的结果不在该曲线上，则它不是椭圆曲线。

椭圆曲线上的点能够相加，但不能相乘。可以定义一种数量乘法(Scalar Multiplication)运算：将一个正整数和椭圆曲线上一个点相乘，简称“数乘”。若 n 是一个正整数，而 P 是椭圆曲线上的一个点，则数量乘法 nP 的运算结果就是对点 P 自身累加 n 次。例如，$2P=P+P$，$5P=P+P+P+P+P$。习惯上，数乘运算也称标量乘法。其中关键的基本步骤是对点 P 加倍的倍点运算，即求解 $2P=P+P$。

倍点运算是点加运算中的特例，即 $P=Q$。这时取 $x_1=x_2$ 代入式(2.24)即可。

需要注意的是，上面的数乘运算定义在正整数域上面，可以通过式(2.25)所定义的法则将数乘运算扩展到整个整数域上。

$$\begin{cases} 0P = O \\ (-n)P = n(-P) \end{cases} \tag{2.25}$$

显然，数乘运算这时实际上是普通 Abel 有限乘法群中的模指数幂运算 g^a 在椭圆曲线群中的体现。

在椭圆曲线中，若对于某一自然数 n，有 $nP=O$，则对于 $m>n$ 有 $mP=(n-m)P$，这反映了椭圆曲线上的数乘运算的一种周期性。

特别地，有下面的定义。

定义2.42　对于椭圆曲线E上的任一点P，若存在最小的正整数m，使得$mP=O$成立，则称m是点P的阶，同时称点P为m的挠点。若存在两个整数a和b，满足$aP=bP$，则a和b满足

$$a \equiv b \bmod m$$

和椭圆曲线群的阶$\#E$一样，它们都是椭圆曲线密码体系中的重要参数。对于定义在有限域$GF(q)$上的椭圆曲线有限群E而言，其上任一点P的阶总是存在的，并且能被椭圆曲线有限群E的阶$\#E(GF(q))$整除(椭圆曲线有限群E的阶是指有限域$GF(q)$上椭圆曲线E中的有理点的总数，包括无穷远点O)。

例如，已知椭圆曲线E的方程是：$y^2=x^3+1$，对于E上的某一点$P=(2,3)$，由式(2.21)和式(2.22)可知：

$$\lambda=\frac{3x_1^2}{2y_1}=\frac{3\times(2)^2}{2\times 3}=2$$

$$v=\frac{-x_1^3+2}{2y_1}=\frac{-2^3+2}{2\times 3}=-1$$

代入式(2.24)，即可得$2P=(0,1)$。同理，可得$3P=(-1,0)$，$4P=(0,-1)$。

显然，$4P=-2P$。所以，$6P=O$，即点P的阶为6。

2.8　自同态环

椭圆曲线的自同态环在椭圆曲线的研究中占有重要地位，是椭圆曲线密码系统的快速实现和解决计算有限域上的椭圆曲线群的阶等问题的核心。

在本节中，始终假设E_1和E_2是两条定义在域F上的椭圆曲

线。

定义 2.43 设 Φ 是由 E_1 到 E_2 的一个有理映射，若 Φ 在 E_1 上的每一点都正则，那么称 Φ 是 E_1 到 E_2 的态射(Morphism)。进一步地，对于态射 Φ，若还满足条件 $\Phi(O)=O$，则称 Φ 是由 E_1 到 E_2 之间的一个同种(Isogeny)。

关于 E_1 和 E_2 之间的同种，有以下一些基本性质。

定理 2.18 设 Φ 是 E_1 到 E_2 之间的一个同种，则当 $\Phi(E_1)\neq O$ 时，一定有 $\Phi(E_1)=E_2$。

当 $\Phi(E_1)=O$ 时，称 Φ 是一个零同种或常数同种。因此，定理 2.18说明，任何同种，要么是一个常数同种(即零同种)，要么一定是满的。下面的定理则说明了同种映射一定是一个同态映射。

定理2.19 设 Φ 是由 E_1 到 E_2 之间的一个同种，则对于任意的 $P,Q\in E_1$，必有

$$\Phi(P+Q)=\Phi(P)+\Phi(Q)$$

定理 2.20 设 Φ 是由 E_1 到 E_2 的一个非常数同种，那么 $\mathrm{Ker}\Phi=\Phi^{-1}(O)$ 一定是一个有限子群。

定义2.44 对于 E_1 到 E_2 的任意两个同种 Φ 和 Ψ，现定义它们的和与积分别为

$$(\Phi+\Psi)(P)=\Phi(P)+\Psi(P) \qquad \text{对任意的 } P\in E_1$$

$$(\Phi\cdot\Psi)(P)=\Phi(\Psi(P)) \qquad \text{对任意的 } P\in E_1$$

那么全部 E_1 到 E_2 之间的同种，连同零同种将构成一个环，它是 E_1 到 E_2 的一个同态环，记作 $\mathrm{Hom}(E_1,E_2)$。

特别地，当 $E_1=E_2=E$ 时，称环 $\mathrm{Hom}(E_1,E_2)$ 为椭圆曲线 E 的自同态环，并记作 $\mathrm{End}(E)$。

环 $\mathrm{End}(E)$ 的单位群，即 E 的全体自同构所组成的子群，称为 E 的自同构群，并记作 $\mathrm{Aut}(E)$。

关于 $\mathrm{Aut}(E)$ 和 $\mathrm{End}(E)$，有下面的结论。

定理2.21　设E是一条定义在域F上的椭圆曲线，那么

(a) E的自同态环End(E)总是下列三种情况之一：

$$\mathrm{End}(E)=\begin{cases}Z\\ \text{一个复二次虚域的阶}\\ \text{一个四元代数的最大阶}\end{cases}$$

且对于其中第二种情况，仅当Char(F)>0时才会发生。

(b) 当Char(F)≠2,3时，

$$\mathrm{Aut}(E)=\begin{cases}\mu_2,\text{若 } j(E)\neq 0,1728\\ \mu_4,\text{若 } j(E)=1728\\ \mu_6,\text{若 } j(E)=0\end{cases}$$

这里，μ_n表示全体n次单位根所构成的群。

当End(E)≠Z时，称椭圆曲线E具有复乘(Complixity Multiplication)。当Char(F)≠0时，一般说来，域F上的曲线都具有复乘。

设$K(E_1)$，$K(E_2)$分别是E_1和E_2的函数域，那么，对于E_1到E_2的同种Φ，可导出一个由$K(E_2)$到$K(E_1)$的一个映射Φ^*如下：

$$\Phi^*:\begin{cases}K(E_2)\to K(E_1)\\ f\to f\cdot\varphi\end{cases}$$

用$\Phi^*K(E_2)$表示Φ^*的象，则显然$\Phi^*K(E_2)\subset K(E_1)$，即$\Phi^*K(E_2)$是$K(E_1)$的子域。

定义2.45　若$K(E_1)$作为$\Phi^*K(E_2)$的扩域时，是可分的、不可分的或者是纯不可分的，则分别称同种Φ是可分的、不可分的或者是纯不可分的，同时，分别称$K(E_1)$为$\Phi^*K(E_2)$的次数、可分次数和纯不可分次数，同时也是同种Φ的次数、可分次数和纯不可分次数，分别用$\deg\Phi$、$\deg_s\Phi$、$\deg_i\Phi$表示。特别地，

$$\deg\Phi=[K(E_1):\Phi^*K(E_2)]$$

定理2.22　设$\Phi:E_1\to E_2$是一个非常数的同种，则对于每一个

$Q \in E_2$，有

$$\#\Phi^{-1}(Q) = \deg_s \Phi$$

特别地，当Φ可分时，

$$\#\Phi^{-1}(O) = \deg\Phi$$

定理2.22指出，对于由E_1到E_2的同种Φ，E_2中每一个元素Q在E_1中逆象的个数是有限的和相同的，且都等于Φ的可分次数，而当Φ可分时，就等于Φ的次数。特别地，可分同种Φ的核中元素的个数就等于Φ的次数。

定理2.23 设E是一条椭圆曲线，C是E的任一有限子群，则一定存在唯一的一条椭圆曲线E'和一个由E到E'的可分同种Φ，使$\mathrm{Ker}\Phi = C$。

定理2.23中的曲线E'一般称为E的同种曲线，有时用E/C表示。

定理2.24 设E是定义在域F上的一条椭圆曲线，S是E的一个有限子群且在F上是稳定的Galois群，则一定存在定义于F上的另一条椭圆曲线E'和唯一的可分同种$\Phi: E \to E'$，使Φ的核$\mathrm{Ker}\Phi = S$。

定理2.24在计算有限域上椭圆曲线群的阶时的作用更为直接。

在求解椭圆曲线数点问题的Schoof算法的改进中，最关键的问题是如何由已知的$E, E', j(E), j(E')$及其$\deg\Phi$等，具体求出Φ的核$\mathrm{Ker}\Phi$的表达式。

定义2.46 设E定义于域F，对于任意正整数m，定义E到自身的同种$[m]$如下：

① 当$m > 0$时，定义$[m]: P \to P + P + \cdots + P = mP \quad (P \in E)$；

② 当$m < 0$时，定义$[m]: P \to (-P) + \cdots + (-P) = mP$ $(P \in E)$。

称这一同种是一个数乘同种$[m]$。

对于数乘同种$[m]$，有下面的性质。

定理2.25　设椭圆曲线E定义在域F上，m为任一整数，则

① $[m]$是一个可分的同种；

② $\deg[m]=m^2$；

③ 用$E[m]$表示同种$[m]$的核，即$E[m]=\{P\in E\mid mP=O\}$。关于$E[m]$的构造，对于任何正整数e，有

① 当$\mathrm{Char}(F)=0$，或F的特征p不被m整除时，$E[m]\cong Z/mZ\oplus Z/mZ$；

② 当$\mathrm{Char}(F)=p\neq 0$时，要么$E[p^e]\cong\{O\}$，要么$E[p^e]\cong\{Z/p^eZ\}$。

定理2.26　设$\Phi:E_1\to E_2$是一个次数$\deg\Phi=m$的非常数同种，则存在唯一的由E_2到E_1的Φ'，使得$\Phi\cdot\Phi'=[m]$。此时，称同种Φ'是Φ的对偶同种。若$\Phi=[0]$，则定义$\Phi'=[0]$。

关于对偶同种，有下面的基本性质。

定理2.27　设$\Phi:E_1\to E_2$是一个同种，记$m=\deg\Phi$，则

① 在E_1上有$\Phi'\cdot\Phi=[m]$，在E_1上有$\Phi\cdot\Phi'=[m]$；

② $\deg\Phi'=\deg\Phi=m$；

③ $\widehat{(\Phi)}=\Phi$。

定理2.28　设$\Phi:E_1\to E_2$，$\Psi:E_1\to E_2$，$\lambda:E_2\to E_3$，是任意的同种，而$m\in Z$是任意的整数，则

① $\widehat{\Phi+\Psi}=\Phi'+\Psi'$；

② $\widehat{\lambda\cdot\Phi}=\Phi'\cdot\lambda'$；

③ $\widehat{[m]}=[m]$，$\deg[m]=m^2$。

第3章　椭圆曲线离散对数

椭圆曲线离散对数问题是椭圆曲线公钥密码学的核心。本章在介绍有限域上的离散椭圆曲线的基础上，讨论椭圆曲线离散对数问题，研究目前已知的椭圆曲线离散对数问题的几类求解算法，分析这些算法的特点和应用范围，最后总结归纳出一系列的安全椭圆曲线选取准则。

3.1　有限域上的离散椭圆曲线

在密码学中，人们关心的只是定义在有限域上的离散椭圆曲线，这种椭圆曲线上的所有点的坐标值均为整数，并且均落在某一个区域内。区域越大，密钥越长，基于这条椭圆曲线的系统就越安全，但系统的计算性能则越低；反之，区域越小，密钥越短，则计算性能越高，但基于这条椭圆曲线的系统的安全性则越低。因此，椭圆曲线密码体系的安全性与高效性之间是一对矛盾，较高的安全性要求必然导致系统计算性能的降低。为了在这对矛盾中取得平衡，在不牺牲安全性的前提下选择合适的区域，保证系统的性能，需要深入了解有限域上的离散椭圆曲线及其群运算法则。

前文提到，与椭圆曲线公钥密码学有关的有限域有三类：$GF(p)$、$GF(2^n)$和$GF(p^m)$。下面将分别讨论这三种有限域上的椭圆曲线。

1. 素数有限域

由前文可知，素数有限域$GF(p)$的特征值为$\mathrm{Char}(F)=p$，其中，p为大于3的素数。由公式(2.14)可知，此时的Weierstrass方程式为

$$y^2 = x^3 + ax + b \qquad (a,b \in F)$$

由前文可知，在讨论椭圆曲线时，约定$\Delta(E)\neq 0$。则对于有限域$GF(p)$，由式(2.11)和式(2.12)可知，a,b满足条件

$$\Delta(E) = 4a^3 + 27b^2 \bmod p \neq 0$$

这时，式(2.19)～式(2.24)所描述的椭圆曲线的群运算法则可以简化为

① 逆元公式：

$$-P = (x, -y) \tag{3.1}$$

② 加法运算：

$$\begin{cases} x_3 \equiv (\lambda^2 - x_1 - x_2) \bmod p \\ y_3 \equiv [\lambda(x_1 - x_3) - y_1] \bmod p \end{cases} \tag{3.2}$$

式中，

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{当 } P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & \text{当 } P = Q \end{cases} \tag{3.3}$$

例如，对于有限域$GF(13)$上的椭圆曲线$y^2=x^3+10x+5$，通过计算可知，其上的点有：$E=\{O,(1,4),(1,9),(3,6),(3,7),(8,5),(8,8),(10,0),(11,4),(11,9)\}$，则该椭圆曲线群的阶$\#E=10$。

若设$P=(1,4)$，$Q=(3,6)$，则

① 求点P的逆元：由式(3.1)可知

$$-P=(1,-4)=(1,9)$$

② 求$P+Q$：

由式(3.3)有

$$\lambda=\frac{6-4}{3-1}=1$$

由式(3.2)有

$$\begin{cases}x_3\equiv(1^2-1-3)\bmod 13=10\\ y_3\equiv(1\times(1-10)-4)\bmod 13=0\end{cases}$$

所以　$P+Q=(10,0)$

③ 求$2P$：

由式(3.3)有

$$\lambda=\frac{3\times 1^2+10}{2\times 4}=\frac{13}{8}\equiv 0 \bmod 13$$

由式(3.2)有

$$\begin{cases}x_3\equiv(0^2-1-1)\bmod 13=11\\ y_3\equiv(0\times(1-11)-4)\bmod 13=9\end{cases}$$

所以　$2P=(11,9)$

2. 二元有限域$GF(2^n)$

二元有限域$GF(2^n)$的特征值为Char$(F)=2$。这时，椭圆曲线非奇异的条件是$\Delta(E)\neq 0$和$j(E)\neq 0$。由公式(2.15)知，此时的Weierstrass方程式为

$$y^2+xy=x^3+ax^2+b\qquad(a,b\in F)$$

且a,b满足条件

$$\Delta(E)=b\neq 0$$

类似地，式(2.19)～式(2.24)所描述的椭圆曲线的群运算法则可以简化为

① 逆元公式：

由于在$GF(2^n)$上有$-n=n$，则$GF(2^n)$上的逆元公式为

$$-P=(x,-x-y)=(x,x+y) \tag{3.4}$$

② 加法运算：

$$x_3=\begin{cases}\lambda^2+\lambda+x_1+x_2+a & \text{当 } P\neq Q\\ \lambda^2+\lambda+a & \text{当 } P=Q\end{cases} \tag{3.5}$$

$$y_3=(x_2+x_3)\lambda+x_3+y_2 \tag{3.6}$$

式中有

$$\lambda=\begin{cases}\dfrac{y_1+y_2}{x_1+x_2} & \text{当 } P\neq Q\\ x_1+\dfrac{y_1}{x_1} & \text{当 } P=Q\end{cases} \tag{3.7}$$

例如，设有限域$GF(2^3)$，其多项式为域多项式$t^3+t+1=0$，现设域上的椭圆曲线E的方程为$y^2+xy=x^3+(t+1)x^2+1$，则其上的点集为

$E=\{O,((000),(001)),((010),(100)),((010),(110)),$
$((011),(100)),((011),(111)),((100),(001)),$
$((100),(101)),((101),(010)),((101),(111)),$
$((110),(000)),((110),(110)),((111),(001)),$
$((111),(110))\}$

该椭圆曲线群的阶$\#E=14$。

若设$P=((010),(100))$，$Q=((000),(001))$，则

① 求点P的逆元。由式(3.4)得

$$-P=((010),(010)+(100))=((010),(110))$$

② 求$P+Q$。由式(3.7)得

$$\lambda=\frac{(100)+(001)}{(010)+(000)}=(101)\times(010)^{-1}$$

$$= (101) \times (101) = (10001) = (111)$$

由式(3.5)和式(3.6)得

$$\begin{cases} x_3 = (111)^2 + (111) + (010) + (000) + (011) = (101) \\ y_3 = ((000) + (101)) \times (111) + (101) + (001) = (010) \end{cases}$$

所以　　$P+Q = ((101),(010))$

③ 求$2P$。由式(3.7)得

$$\lambda = (010) + \frac{(100)}{(010)} = (010) + (100) \times (010)^{-1}$$

$$= (010) + (100) \times (101) = (10110) = (000)$$

由式(3.5)和式(3.6)可得

$$\begin{cases} x_3 = (000)^2 + (000) + (011) = (011) \\ y_3 = ((010) + (011)) \times (000) + (011) + (100) = (111) \end{cases}$$

故　　$2P = ((011),(111))$

3. 素特征扩张有限域$GF(p^m)$

由于$GF(p^m)$的特征值为$\mathrm{Char}(F)=p$，所以，其群运算法则与$GF(p)$相同。

表3.1　有限域上的椭圆曲线

素数p	7	97	997
素域$GF(p)$上可用椭圆曲线的数量(条)	30	9 181	992 020

对于一个确定的有限域而言，其上的椭圆曲线资源非常丰富。从表3.1可以看出，当素数p增加时，有限域$GF(p)$上的可用椭圆曲线的数量迅速增加，其增长速度远远超过素数p的增长速度。当$p=997$时，素域$GF(p)$上的可用椭圆曲线的数量已经达到992 020条。而对于实际应用中的椭圆曲线密码体系(p为不少于160位的大素数)而言，其上的椭圆曲线数量更是惊人。因而，可供

选取的椭圆曲线相当丰富，这为椭圆曲线密码体系的设计带来了很大的方便，对于寻找合适的、安全的椭圆曲线是非常有利的。

椭圆曲线密码体系的安全性与有限域上的椭圆曲线自身的安全性密切相关，一个高安全性的椭圆曲线密码体系可以通过选择安全的椭圆曲线来获得。而如何确定一条椭圆曲线的安全性，则依赖于椭圆曲线离散对数问题的求解困难性。

3.2 椭圆曲线离散对数问题

自 1976 年，公钥密码体系被发现以来，任何一个公钥密码体系都是以某一含有“陷门”的数学难题作为其安全基础的。Odlyzko 指出，就椭圆曲线公钥密码体系而言，各种椭圆曲线公钥密码体系的安全性都与相应的椭圆曲线离散对数问题的求解困难性等价。本书认可这一观点，以下约定各种椭圆曲线公钥密码体系的安全性都与相应的椭圆曲线离散对数问题的求解困难性等价，将椭圆曲线公钥密码体系的安全性归结为对单纯的椭圆曲线离散对数问题的求解，从一般意义上的离散对数问题入手，着重讨论椭圆曲线离散对数问题。

一般地，一个群 G 上的离散对数问题定义如下。

定义3.1 设 G 是任一有限Abel 加法群，$P,Q\in G$ 为 G 的任意两群元。若已知存在整数 m，使得 $Q=P^m$ 成立，其中 P^m 由定义 2.9 所定义，表示对群元 P 进行 m 次模运算，则由群元 P,Q 及群 G 求出 m 的问题称为群 G 上的离散对数问题，简记为 DLP (Discrete Logarithm Problem)，而称数 m 为群元 Q 的离散对数，同时称群元 P 为生成元。

显然，对于有限Abel 群 G 上的任何DLP，都能在 $O(|G|)$ 的时

间内通过穷举搜索逐一计算$P,P^2,P^3,\cdots,P^m$,得到求解。而在实际中,当群G的阶足够大时,穷举搜索是没有实际意义的。为此,需要定义有关DLP问题的求解困难性,其概念如下。

定义3.2 设G是一有限Abel群,A是求解G上任何一个DLP问题的一种算法。当算法A的时间复杂度为$O(\ln|G|)$时,称算法A是求解G上DLP问题的一个多项式算法,并称G上的DLP问题是可解的。相应地,对于G上DLP,若不存在求解它们的多项式算法,则称群G上DLP求解是困难的,这时,当G的阶足够大时,求解该问题在计算上是不可行的。

由于定义在有限域上的离散椭圆曲线群是一个有限Abel加法群,其基本运算是椭圆曲线群上的群加法运算,即点加运算。相应地,群运算P^m相当于椭圆群上的数乘运算。因而可以得到以下类似于定义3.1的椭圆曲线离散对数问题的定义。

定义3.3 用$E(F)$表示定义在有限域$GF(q)$上的椭圆曲线E在扩域F上的有理子群,设任意两点$P,Q\in E(F)$,若已知对某整数m有$Q=mP$成立,则称数m为点Q的椭圆曲线离散对数,简记为ECDL(Elliptic Curve Discrete Logarithm);而由P,Q及E求出m的问题则称为E上的椭圆曲线离散对数问题,简记为ECDLP(Elliptic Curve Discrete Logarithm Problem);相对于点Q,点P称为基点(Base Point)。

显然,椭圆曲线离散对数是不唯一的。设点P的阶为r,且整数m,n均为E的椭圆曲线离散对数,则m,n之间满足$m\equiv n \bmod r$。故在椭圆曲线离散对数问题中,一般约定椭圆曲线离散对数m是一个小于r的正整数。

类似于定义3.2,有以下关于椭圆曲线离散对数问题困难性的定义。

定义3.4 对于定义在有限域$GF(q)$上的椭圆曲线E,设A是

求解G上任何一个ECDLP问题的一种算法。当算法A的时间复杂度为$O(\ln|E|)$时，称算法A是求解E上ECDLP问题的一个多项式算法，并称E上的ECDLP问题是可解的。相应地，对于E上ECDLP，若不存在求解它们的多项式算法时，则称E上ECDLP求解是困难的，或者说，该ECDLP问题在计算上是不可解的。

与一般的有限乘法群上的离散对数问题不同，有限域上的椭圆曲线离散对数问题的求解更难。这是因为在一般的DLP问题中，有限域上的代数对象由域加法和域乘法两种基本运算构成，这使得亚指数时间的DLP指标积分算法可行；而在ECDLP问题中，其代数对象只包括一种基本运算，即椭圆曲线群上的点加运算，这使得除了少数非常特殊的，可以转化为有限乘法域的椭圆曲线以外，亚指数时间的DLP求解算法对ECDLP问题无效。目前，针对一般ECDLP问题，尚未出现好的低于指数级时间的求解算法。

目前，针对ECDLP的求解算法主要有以下三类。

① 针对一般DLP问题的大步小步算法和Pollard-ρ算法等。

② 针对超奇异型(Supersingular)椭圆曲线的MOV类演化算法。

③ 针对畸型(Anomolus)椭圆曲线的SSAS多项式时间算法。

椭圆曲线密码体系的安全性是由ECDLP问题求解的困难性决定的，因而通过分析现有的各种ECDLP求解算法，准确把握ECDLP问题求解进展情况，对于设计、评价快速安全的椭圆曲线密码体系具有非常重要的意义。

在随后的几节中，将分别研究上述的几种ECDLP求解算法，最后，给出安全椭圆曲线的判定准则，归纳出安全椭圆曲线的类型。

3.3 一般椭圆曲线上的离散对数问题的求解

本节所要讨论的大步小步算法、Pohlig-Hellman 算法、Pollard-ρ 算法和 Xendi 算法等都是针对一般的 N 阶有限 Abel 群上的离散对数问题 DLP 的求解算法。

设 G 是一个 N 阶有限 Abel 加法群，G 中的群运算为加法。群元 $P,Q\in G$，整数 $m\in[0,N]$，记 $Q=mP$，则离散对数问题要求从 P、Q、N 及 G 等已知条件中求出 m。1978 年以前，求解 m 的最好算法是 Shanks 于 1972 年提出的大步小步(Baby Step-Giant Step)算法。1978 年，Pohlig 和 Hellman 提出了将求解 m 的问题演化成为重点考虑对群 G 的素数阶循环子群上一般离散对数问题的 Pohlig-Hellman 演化类求解算法。对于一般的素数阶循环群上的离散对数问题，Pollard 于 1978 年在大步小步算法的研究基础上，给出了一种基于大步小步算法的概率求解算法，即 Pollard-ρ 算法，它实际上是大步小步算法的改进变形版本。虽然该算法的时间复杂度与 Shanks 的大步小步算法的时间复杂度相当，但其空间复杂度却可以忽略不计。之后，Van Oorschot 和 Wiener 在 1998 年提出了将 Pollard-ρ 算法分成 m 个过程进行并行化处理，成为分布式 Pollard-ρ 算法，通常称为 Pollard-λ 算法。该算法是目前已知对一般 ECDLP 问题的最快的求解算法。

本节将在讨论这几种算法的基础上，给出它们在求解椭圆曲线离散对数问题上的推广、改进和变形，以说明一般椭圆曲线离散对数问题的求解难度。

3.3.1　大步小步算法

这一算法的思想仍是基于穷举搜索算法，它采用“以空间换取时间”的策略，将穷举搜索中的部分时间换为空间存储。该算法的工作原理如下。

为了由P、Q、N及G等已知条件中求出离散对数问题$Q=mP$中的m，首先令$n_1=[\sqrt{N}]$，则对于任何$l\in[0,N]$，l可被唯一地表示为

$$l = an_1 + b$$

式中，$0\leqslant a,b\leqslant n_1$。这样，对于$Q=mP$中的$m$，必然存在唯一的$a_0$，$b_0\in[0,n_1]$，使得

$$m = a_0 n_1 + b_0 \tag{3.8}$$

将式(3.8)代入$Q=mP$，有

$$Q = a_0 n_1 P + b_0 P$$

即

$$Q - a_0(n_1 P) = b_0 P$$

记$P_1=n_1P$，则有

$$Q - a_0 P_1 = b_0 P \tag{3.9}$$

其中，$n_1=[\sqrt{N}]$是已知的，所以$P_1=n_1P$已知，且有$a_0,b_0\in[0,n_1]$，所以，可在区间$[0,n_1]$上用穷举搜索法求解离散对数问题。

大步小步算法的具体步骤如下。

第一步(小步)：对于$b=0,1,2,\cdots,n_1$，计算bP，将结果排序后存储起来。

第二步(大步)：依次对$a=0,1,2,\cdots,n_1$，计算$Q-aP_1$，并在已

存储的bP列表中查找。若对于某一a值(记为a^*),有$Q-a^*P_1$与表中某个b值(记为b^*)对应的b^*P相同,即

$$Q - a^*P_1 = b^*P$$

则由a_0,b_0的唯一性可知,$a_0=a^*$,$b_0=b^*$,按式(3.8)即可求出m。

大步小步算法的时间复杂度是$O(\sqrt{N})$,空间复杂度也是$O(\sqrt{N})$,所以,该算法实际上是对穷举搜索策略在时间和空间上的一种折中。

3.3.2 Pohlig-Hellman 演化类算法

首先介绍一下著名的中国古代剩余定理(又称孙子定理)。

定理 3.1 (中国古代剩余定理)设$m_1,m_2,\cdots,m_r$是两两互素的r个整数,$a_1,a_2,\cdots,a_r$是任意r个整数,设有如方程组(3.10)所示的由r个同余方程$x\equiv a_i \bmod m_i$组成的同余方程组。

$$\begin{cases} x \equiv a_1 \bmod m_1 \\ x \equiv a_2 \bmod m_2 \\ \vdots \\ x \equiv a_r \bmod m_r \end{cases} \tag{3.10}$$

则方程组(3.10)的模$M=m_1m_2\cdots m_r$有唯一解。该解可由式(3.11)得出。

$$x = \sum_{i=1}^{r} a_i M_i y_i \mod M \tag{3.11}$$

式中,$M_i=M/m_i$,$y_i=M_i^{-1} \bmod m_i$,$1\leqslant i\leqslant r$。

定理3.1所描述的中国古代剩余定理指出了如何由剩余类构造模M域的方法。

设群元P的阶为n,并设n的标准因子分解式为

$$n = p_1^{f_1} p_2^{f_2} p_3^{f_3} \cdots p_r^{f_r} \tag{3.12}$$

若对于每一个 $p_i^{f_i}(i=1,2,\cdots,r)$都已经求出了 $m \bmod p_i^{f_i}$，则由中国古代剩余定理3.1可立即求出m。据此，Pohlig-Hellman 演化算法的基本思路如下。

首先，由N的素因子分解式求出n的素因子分解式(3.12)；然后，对于每一个$i=1,2,\cdots,r$，求出 $m \bmod p_i^{f_i}$；最后，再由中国古代剩余定理3.1求出离散对数问题的解m。

为了求出 $m \bmod p_i^{f_i}$，可将一般离散对数问题$Q=mP$转化为对新离散问题$Q'=m'P'$的求解，其中，P'的阶是p_i，且m'满足$1 \leqslant m' \leqslant p_i$。

下面，给出Pohlig-Hellman 演化算法的具体工作过程。

首先，设群G的阶N的标准素因子分解式为

$$N = p_1^{e_1} p_2^{e_2} p_3^{e_3} \cdots p_r^{e_r} \tag{3.13}$$

式中，$e_i \geqslant 1$。

因为群元P的阶n能被群G的阶N整除，即$n \mid N$，所以由n的素因子分解式(3.12)，有$0 \leqslant f_i \leqslant e_i, i=1,2,\cdots,r$。

由于n是群元P的阶，故$nP=0$，所以$NP=0$。现在考虑式$\left(\frac{N}{p_i}\right)P$：当$\left(\frac{N}{p_i}\right)P \neq O$时，一定有$f_i=e_i$；而当$\left(\frac{N}{p_i}\right)P=O$时，则需要继续判定$\left(\frac{N}{p_i^2}\right)P$的值，如此下去，在$e_i$步内一定能够确定$f_i$ $(i=1,2,\cdots,r)$。这样，就实现了从N的标准素因子分解式(3.13)得到n的素因子分解式(3.12)的目标。

现在，对于每一个$i=1,2,\cdots,r$，考虑求 $m \bmod p_i^{f_i}$ 的值。为简化符号，下面用p表示这里的p_i，用f表示这里的f_i，并设$f \geqslant 1$。

记$m_p = m \bmod p^f$，则存在正整数k，使得

$$m = m_p + k \times p^f \tag{3.14}$$

设

$$m_p = \sum_{i=0}^{f-1} a_i p^i = a_0 + a_1 p + a_2 p^2 + \cdots + a_{f-1} p^{f-1}$$

$$(0 \leqslant a_i < p) \qquad (3.15)$$

注意到$nP=O$，$f \geqslant 1$，则由式(3.14)，对于每一个$k=1,2,\cdots,f$，有

$$\begin{aligned} m\left(\frac{n}{p^k}\right)P &= \left(\frac{m}{p^k}\right)nP = \left(\frac{m_p + k \times p^f}{p^k}\right)nP \\ &= \left(\frac{m_p}{p^k}\right)nP + \left(\frac{kp^f}{p^k}\right) \times nP \\ &= \left(\frac{m_p}{p^k}\right)nP + kp^{f-k} \times np \\ &= m_p\left(\frac{n}{p^k}\right)P \end{aligned}$$

即

$$m\left(\frac{n}{p^k}\right)P = m_p\left(\frac{n}{p^k}\right)P \qquad (3.16)$$

再进一步，由式(3.15)，有

$$\begin{aligned} m_p\left(\frac{n}{p^k}\right)P &= \left(\frac{m_p}{p^k}\right)nP = \left(\frac{\sum_{i=0}^{f-1} a_i p^i}{p^k}\right)nP \\ &= \left(\frac{\sum_{i=0}^{k-1} a_i p^i}{p^k}\right)nP + \left(\sum_{i=k}^{f-1} a_i p^{i-k}\right)nP \\ &= \sum_{i=0}^{k-1} a_i\left(\frac{n}{p^{k-i}}\right)P \end{aligned}$$

从而

$$m_p\left(\frac{n}{p^k}\right)P = \sum_{i=0}^{k-1} a_i\left(\frac{n}{p^{k-i}}\right)P \qquad (3.17)$$

结合式(3.16),有

$$m\left(\frac{n}{p^k}\right)P=\sum_{i=0}^{k-1}a_i\left(\frac{n}{p^{k-i}}\right)P \tag{3.18}$$

对方程$Q=mP$的两边同时乘以$\left(\frac{n}{p^k}\right)$后,由式(3.18),有

$$\begin{aligned}\left(\frac{n}{p^k}\right)Q&=\left(\frac{n}{p^k}\right)mP=m\left(\frac{n}{p^k}\right)P\\&=\sum_{i=0}^{k-1}a_i\left(\frac{n}{p^{k-i}}\right)P\end{aligned} \tag{3.19}$$

当$k=1$时,式(3.19)简化为

$$\left(\frac{n}{p}\right)Q=\left(\frac{n}{p}\right)mP=a_0\left(\frac{n}{p}\right)P$$

记$Q_1=\left(\frac{n}{p}\right)Q$,$P_1=\left(\frac{n}{p}\right)P$,则可得

$$Q_1=a_0P_1 \tag{3.20}$$

式中,$0\leqslant a_0<p$。

对式(3.20),可利用大步小步算法,在$O(\sqrt{p})$步内求出a_0。

对$k=2,\cdots,f$,设$a_i(i=0,\cdots,k-2)$已知,则由式(3.19),有

$$\left(\frac{n}{p^k}\right)Q-\sum_{i=0}^{k-2}a_i\left(\frac{n}{p^{k-i}}\right)P=a_{k-1}\left(\frac{n}{p}\right)P \tag{3.21}$$

令

$$\begin{cases}Q_k=\left(\frac{n}{p^k}\right)Q-\sum_{i=0}^{k-2}a_i\left(\frac{n}{p^{k-i}}\right)P\\P_k=\left(\frac{n}{p}\right)P=P_1\end{cases}$$

则式(3.21)可化简为

$$Q_k=a_{k-1}P_k \tag{3.22}$$

式中,对于$0\leqslant a_i<p$,$i=1,\cdots,f-1$。

对于离散对数问题(3.22),依次应用大步小步算法,在

$O(\sqrt{p})$步内求出a_{k-1}。这样，在f步内，可以依次求解出所有的a_0，$a_1,\cdots,a_{f-1}$，然后由式(3.15)可以求出m_p。

对于式(3.12)中所有的素数p_i $(i=1,2,\cdots,r)$，当求出所有的$m \pmod{p_i^{f_i}}$后，即可利用中国古代剩余定理3.1，最后求出离散对数m。

Pohlig-Hellman 算法实际上是一种演化算法，它的最大功能是将N阶有限群G上的离散对数问题演化为N的所有素因子的循环子群上的若干离散对数问题，从而加速了群G上离散对数问题的求解过程。当基点的阶n的因子全部是小素数因子时，这种算法是非常有效的。

因此，从密码学的角度看，为了提高有限群G上离散对数问题的困难性，要求生成元P的阶n中至少包含一个大素数因子，一般地，要求n为大素数。进一步地，对有限群G而言，要求其阶N中应该至少包含一个尽可能大的素数因子。在群阶N一定的情况下，要使N中包含的素数因子尽可能的大，则N中的素数因子的个数不能太多，更不能有两个或更多的较大的素数因子，最好是N中只有一个大素数因子。当然，最理想的情况是群阶N本身就是一个大素数。因此，在密码系统中，构造离散对数问题时，通常选择素数阶的有限循环群作为基群G，以避免Pohlig-Hellman 算法的攻击。

3.3.3 Pollard-ρ 概率类算法

对于一般的素数阶有限循环群G上的离散对数问题，Pollard于1978年提出了一种基于大步小步算法的概率求解算法。该算法的时间复杂度的期望值是$O(\sqrt{N})$，与Shanks 的大步小步算法相

当，但由于其空间复杂度仅为$O(1)$，所以一般公认Pollard的这一算法优于Shanks的大步小步算法。下面，先对Pollard的这一算法作一个简单的介绍。

由Pohlig-Hellman算法，只需要研究素数阶有限群G上的离散对数问题，这里，不妨假定有限群G的阶是一个素数。

现在首先按照某种简单的规则将有限群G上三个所包含的元素数目大致分为相等的子集S_1,S_2,S_3，然后定义群G上的一个迭代函数$f(R)$　$(R\in G)$如下。

$$f(R)=\begin{cases}2R, & \text{若 } R\in S_1\\ R+P, & \text{若 } R\in S_2\\ R+Q, & \text{若 } R\in S_3\end{cases} \tag{3.23}$$

然后，随机选取整数$A_0,B_0\in[1,N]$，计算函数f的起始群元$R_0=A_0P+B_0Q$，并计算：

$$\begin{aligned}R_1&=f(R_0)\\ R_2&=f(R_1)\\ &\cdots\\ R_i&=f(R_{i-1})\end{aligned} \tag{3.24}$$

显然，对于每一个R_i，都有

$$R_i=A_iP+B_iQ \tag{3.25}$$

且有$A_i\leqslant A_{i+1},B_i\leqslant B_{i+1}$。

现用(R_i,A_i,B_i)表示式(3.25)，并将每次迭代中的(R_i,A_i,B_i)记录下来。这样，所有的(R_i,A_i,B_i)将形成一个长串。若对于某第k次迭代中的R_k，恰好与前面第j次迭代中的R_j相同，即$R_k=R_j$，则由式(3.25)，有

$$A_kP+B_kQ=A_jP+B_jQ$$

故

$$Q = \frac{A_j - A_k}{B_k - B_j} P$$

由于生成元 P 的阶为 N，据此可得

$$m \equiv \frac{A_j - A_k}{B_k - B_j} \bmod N \tag{3.26}$$

由于在得到 $R_k = R_j (j < k)$ 时，全部的迭代结果 (R_i, A_i, B_i) 所摆成的串的形状与字母“ρ”的形状相似，故该算法又得名“ρ”方法，即 Pollard-ρ 算法。

实际上，由迭代函数 f 所得到的群元 $R_1, R_2, \cdots$ 可以看作是从有限群 G 中随机选出的群元。由于 G 中的群元数目是有限的，所以这一迭代过程最终会出现重复。用 M 表示迭代 k 次后仍未出现重复群元的事件，则 $R_1, R_2, \cdots R_k$ 是 G 中的 k 个不同的群元，于是事件 M 的发生概率为

$$\left(1 - \frac{1}{N}\right)\left(1 - \frac{2}{N}\right)\cdots\left(1 - \frac{k-1}{N}\right) \approx e^{\frac{k^2}{2N}}$$

由此，能够得到在 $R_0, R_1, \cdots R_k$ 中存在重复群元时 k 的期望值是 $\sqrt{\frac{\pi N}{2}}$，即 Pollard-ρ 算法的时间复杂度是 $O\left(\sqrt{\frac{\pi N}{2}}\right) \approx O(\sqrt{N})$，与 Shanks 的大步小步算法的时间复杂度基本相同。

关于当 $R_0, R_1, \cdots R_k$ 中存在重复点时，如何在不占用太多的存储空间的情况下，有效地将其检测出来，Brent，Folyd，Frey 和 Sedgewick 等人均做了不少工作，使得 Pollard-ρ 算法能够在几乎不占用多少空间的情况下在 $O(\sqrt{N})$ 的时间内求解离散对数 m。

下面将结合椭圆曲线离散对数问题，综合现有的对 Pollard-ρ 算法的改进研究进展情况，给出重复群元的检测算法。

当 G 是椭圆曲线有限群时，考虑到椭圆曲线有限群中求逆运算几乎不费任何时间的这一特点，在应用 Pollard-ρ 算法求解椭圆

曲线离散对数问题时，能够少量地加速求解过程。

设 E 是定义于有限域 $GF(q)$ 上的椭圆曲线，$P\in E(GF(q))$，椭圆曲线离散对数问题 $Q=mP$ 定义于由基点 P 生成的循环子群 $\langle P\rangle$ 上。

① 对于 $\langle P\rangle$ 中任一点 R，设 $R=(x,y)$，$x,y\in GF(q)$。用 $W(x)$ 表示有限域 $GF(q)$ 中元素 x 用{0,1}序列表示后所直接对应的整数，即整数 $W(x)$ 的二进制表示就是域 $GF(q)$ 中元素 x 用{0,1}序列表示的结果。利用 $W(x)\bmod 3$ 可将 $\langle P\rangle$ 划分为如下三个不同的子集 S_1,S_2 和 S_3：

$$S_1=\{P\in\langle P\rangle\mid W(x)\bmod 3=1\},$$
$$S_2=\{P\in\langle P\rangle\mid W(x)\bmod 3=2\},$$
$$S_3=\{P\in\langle P\rangle\mid W(x)\bmod 3=0\}。$$

② 对于任一 $R\in\langle P\rangle$，定义 $\langle P\rangle$ 上的如式(3.23)所示的迭代函数 f 如下：

$$f(R)=\begin{cases}2R, & 若\ R\in S_1\\ R+P, & 若\ R\in S_2\\ R+Q, & 若\ R\in S_3\end{cases}$$

③ 随机选取整数 $A_0,B_0\in[1,n-1]$，这里，n 表示群 $\langle P\rangle$ 的阶，即 $|\langle P\rangle|=n$。接着计算函数 f 的初始点 $R_0=A_0P+B_0Q$。然后利用迭代函数 f 按式(3.24)计算 R_i。一般地，如式(3.25)所示，当 $R_i=A_iP+B_iQ$ 时，有

$$(R_{i+1},A_{i+1},B_{i+1})=\begin{cases}(2R_i,2A_i,2B_i), & 若\ R\in S_1\\ (R_i+P,A_i+1,B_i), & 若\ R\in S_2\\ (R_i+Q,A_i,B_i+1), & 若\ R\in S_3\end{cases}$$

由此，可得Pollard-ρ序列 $R_0,R_1,R_2,\cdots$。

④ 为了能在序列 $R_0,R_1,R_2,\cdots$ 中即时检测出重复点，采用下列方法。对于每一个 $i=1,2,3,\cdots$，同时计算 (R_i,A_i,B_i) 和 $(R_{2i},$

A_{2i}, B_{2i})，直到对于某一个$i=r$，有$R_r=R_{2r}$为止。这时，有

$$A_rP + B_rQ = A_{2r}P + B_{2r}Q$$

故

$$m \equiv \frac{A_{2r} - A_r}{B_r - B_{2r}} \bmod n$$

这是因为对每一个群元$R\in\langle P\rangle(R\neq O)$，$R$ 都有一个逆元$-R\in\langle P\rangle$。所以，对$R=(x,y)\in\langle P\rangle$和逆元$-R=(x,-y)\in\langle P\rangle$，可以通过其$y$ 坐标来区分群元R 和逆元$-R$。设存在一种简单的映射函数能使域$GF(q)$中元素与正整数一一对应。这时，在y 与$-y$ 中，把较小的那一个重新记为y，另一个记为$-y$。这样，$\langle P\rangle$中的所有非零元素可以被分成两个具有相等阶的子集W 和$-W$。其中，对任何$R\in W$，有$-R\in -W$，并且当$R=(x,y)\in W$ 时，y 所对应的正整数小于$-y$ 所对应的正整数。

在对$\langle P\rangle$中元素按上述方案划分后，当计算序列$R_1, R_2, \cdots$时，稍作调整，可以很容易地使R_i 始终保持在W 中。由于$|W|=\frac{1}{2}(n-1)$，所以，对调整后的序列$R_1, R_2, \cdots, R_i, \cdots$，第一个重复点出现时$i$ 的期望值为$O\left(\sqrt{\frac{\pi n}{4}}\right)$。这一复杂度估计是目前对一般椭圆曲线离散对数问题进行求解时的最佳估计。

1998 年，Van Oorschot 和 Wiener 提出将 Pollard-ρ 算法分成m 个过程进行并行化处理，成为分布式 Pollard-ρ 算法，即 Pollard-λ 算法。该算法的运行时间复杂度估计为$O\left(\sqrt{\frac{\pi n}{2m}}\right)$，是目前已知的对一般ECDLP 问题的最快的求解算法。特别地，对定义在二次有限域$GF(2^r)$的子域$GF(2^l)$上的椭圆曲线离散对数问题，Wiener 和 Zucherato 指出，利用分布式 Pollard-ρ 算法可以将运行时间复杂

度估计减少到$O\left(\frac{1}{2m}\sqrt{\frac{\pi nl}{4m^2r}}\right)$。

此外，Wiener 和 Zucherato 还给出了利用 Pollard-ρ 算法求解子域曲线上的椭圆曲线离散对数问题的方法。特别地，对于定义在二次有限域$GF(2^m)$上的形如$y^2+xy=x^3+x^2+1$的 Koblitz 型椭圆曲线，利用 Pollard-ρ 算法求解基于其上的椭圆曲线离散对数问题时，最好的时间复杂度估计是$O\left(\sqrt{\frac{\pi n}{4m}}\right)$。

3.3.4　Index 算法和 Xedni 算法

对于基于有限乘法群$GF(q)^*$（即除去零元素后的由整数模余构成的交换群）上的离散对数问题，有一种算法能在亚指数时间复杂度内进行求解，这一算法就是 Index Calculus 指标积分算法，简称 Index 算法，该算法的平均时间复杂度为$O(e^{\frac{1}{2}\sqrt{\ln p \ln\ln p}})$，最好情况下可达$O(e^{1.9229+O(1)(\ln p)^{\frac{1}{3}}(\ln(\ln p))^{\frac{2}{3}}})$。

Index 算法最早在1920年被发现，之后又多次被重新发现。该算法使得基于有限乘法群$GF(q)^*$上的离散对数问题的密码体系受到严重威胁，这就迫使人们提出了能否用其他有限群来代替有限乘法群$GF(q)^*$的问题。基于此，1985年，Miller 和 Koblitz 各自独立地指出，对于基于有限域$GF(q)$上的椭圆曲线有限群$E(GF(q))$，由于其上的代数运算只有“点加运算”这种基本运算，所以 Index 算法不能用于求解有限群$E(GF(q))$上的离散对数问题。为此他们提出，可以用椭圆曲线有限群$E(GF(q))$来代替有限乘法群$GF(q)^*$，所生成的密码体系应该拥有更好的安全性，或者说，在同等安全性要求下，可以降低密钥的长度，这导致了椭圆曲

线密码体系的诞生。

在椭圆曲线密码体系诞生后，人们又从攻击椭圆曲线离散对数问题ECDLP的角度，希望能够从Index算法为什么不能用于求解椭圆曲线离散对数问题的原因中，重新考虑如何改进Index算法，构造新的能够求解椭圆曲线离散对数问题的算法。有许多学者在这一方面做了大量的研究工作，其中成绩最卓著的当属J. Silverman。1998年，Silverman在Miller和Koblitz等人工作的基础上，详细分析了Index算法不能用于求解椭圆曲线离散对数问题的原因，并指出：Index算法能够成功的根本原因是有限乘法群$GF(q)^*$中的一些元素a_j能够用另外一些"不太大"的素数元素很容易地线性表示出来。但对于有限域$GF(q)$上的椭圆曲线有限群$E(GF(q))$，这一性质是不存在的，即：

① $E(GF(q))$中很少有"不太大"的元素，这里的"不太大"是指对于群元点$P=(x,y)\in E(GF(q))$，x和y作为$GF(q)$中的元素都不大。

② 假如认为在$GF(q)$中的不同素数之间存在"线性独立"关系的话，则$GF(q)$中同时应包含丰富的这样的元素(即素数)。但在$E(GF(q))$中，是不存在很多"线性独立"元素的。换句话说，对一般的椭圆曲线而言，它的秩都比较小，目前所找到的有理数域Q上的椭圆曲线的最大秩仅为23。

③ 对有限乘法群$GF(q)^*$中的一些元素a_j，总能判断它们是否存在形如$a_j=p_1^{e_{j1}}p_2^{e_{j2}}\cdots p_r^{e_{jr}}$的标准分解式；当存在这种标准分解式时，也能给出这一分解式。但对于$E(GF(q))$中的元素，要将某个元素用另外一些元素"线性表出"是非常困难的。

正是上述这些原因导致了Index算法的思想不能应用到$E(GF(q))$上的椭圆曲线离散对数问题的求解上。

1998年，在加拿大滑铁卢大学(University of Waterloo)举行

的第二届椭圆曲线密码学国际研讨会上，J. Silverman 宣布了一种求解 $E(GF(q))$ 上的椭圆曲线离散对数问题的新的攻击算法，即 Xedni 算法。之所以这样命名，是因为这一算法是将指标积分 Index 算法的顺序倒过来考虑而得到的。

Xedni 算法对椭圆曲线上的点引入了与有限乘法域上分解小素数因子的积相当的概念，把有限域上的椭圆曲线提升为有理数域上的椭圆曲线，利用有理数域上的椭圆曲线的点的高度来求解。

这一算法宣布后曾经引起一定的轰动，但由于这一算法涉及高深的数学理论和复杂的数学计算，使得人们一时还无法判定它的可行性。为此，滑铁卢大学密码研究中心的几个研究小组对该算法的可行性从理论和实验两个方面做了考证，指出：Xedni 算法成功求解椭圆曲线离散对数问题的概率非常小，且当其成功时，其计算量仍是指数时间。主要理由如下。

① 由椭圆曲线上的点的高度的特征可知，被提升点中若存在线性相关关系，则这些相关系数的大小会被一个绝对大小的常数所限制。这意味着随机选取提升点时，可能需要进行 $O(p)$ 次操作才会得到一个线性相关关系。

② 即使被提升的这些点中存在线性相关关系，但在 p 达到实际应用大小，如 $O(2^{160})$ 时，因 $\Delta(E)$ 已经变得非常大，这时，这种线性相关关系出现的概率会变得非常小。

③ 虽然从理论上说，线性相关关系的相关系数应该是丰富的，但在实际中找出这些相关系数却较困难。

④ 虽然 Xedni 算法中所采用的反-Mestre 条件可能会增加线性相关关系出现的概率，但它同时也导致了 $\Delta(E)$ 的增加，因而总体效应可能并不存在。

所以，Xedni 算法是不可行的。

3.4 特殊椭圆曲线上的离散对数问题的求解

由上一节的讨论可知，一般椭圆曲线上的ECDLP 问题的求解算法仍然需要指数级的时间复杂度。但目前发现有两类特殊的椭圆曲线，即“超奇异型”椭圆曲线和“畸形”椭圆曲线，对于基于这些特殊的椭圆曲线离散对数问题，已经找到了有效的求解算法。

本节通过研究这些针对特殊的椭圆曲线离散对数问题的求解算法，指出了在实际设计椭圆曲线密码体系时，为保证系统的安全性，必须避免选用这些特殊的椭圆曲线。

1. MOV 方法

1991 年，Menezes、Okamoto 和 Vanstone 给出了一种通过将有限域$GF(q)$上的椭圆曲线离散对数问题通过“Weil-配对”，演化归约到有限域$GF(q)$的一个扩域$GF(q^t)$上的普通离散对数问题，通过对有限域$GF(q^t)$上的离散对数问题的求解，最终实现对原来的有限域$GF(q)$上的椭圆曲线离散对数问题求解的算法，并以三位作者姓名的首字母将该算法命名为MOV 演化算法，简称MOV 方法。该算法仅对某些特殊类型的椭圆曲线有效，使得对基于这些特殊类型的椭圆曲线有限群上的椭圆曲线离散对数问题，具有亚指数攻击时间。

现设E是定义在特征为p的有限域$GF(q)$上的椭圆曲线，$P\in E(GF(q))$为椭圆曲线E上的一点，且由点P生成的循环子群$\langle P\rangle$的阶是一个不等于p的素数n。现记椭圆曲线E上的n挠点（即n倍点为无穷远点O的点）的集合为$E[n]=\{P\in E\mid nP=O\}$。由

Pholig-Hellman 方法，假定基于椭圆曲线 E 上的椭圆曲线离散对数问题是定义于$\langle P\rangle$上的，即 $P,Q\in\langle P\rangle$已知，$\#E(GF(q))$和 n 等也已知，现要求一整数 $l\in(0,n)$，使得

$$Q = lP \tag{3.27}$$

下面，通过应用除子理论，引入“Weil-配对”的概念，给出 MOV 演化算法的基本思路，并对MOV 演化算法的实现和应用条件进行简要地分析。

定理3.2　设椭圆曲线 E 定义在域 F 上，$\overline{F}(E)$是 E 的函数域，$D=\sum n_p(P)$ 是 E 的任一除子，则 D 是主除子(即存在 $f\in\overline{F}(E)$)，使得 $D=\mathrm{div}(f)$的充分必要条件为

(1) $\sum n_p = 0$;

(2) $\sum n_pP = O$。

定理3.3　设椭圆曲线 E 定义在域 $GF(q)$上，n 是不同于 p 的一个素数，且 n 不能整除 $\#E(GF(q))$，n 也不能整除$(q-1)$，则当且仅当 n 不能整除(q^t-1)时，$E[n]\subset E(GF(q^t))$。

现对满足 $\gcd(n,p)=1$ 的已知整数 n，设 $T\in E[n]$，对除子

$$D = n(T) - n(O)$$

由定理3.2，存在 $f\in\overline{F}(E)$，使得

$$\mathrm{div}(f) = n(T) - n(O)$$

对这里的 T，设 T'是 E 中使 $nT'=T$ 的另外一点。对 T'，令除子

$$D' = \sum_{R\in E[n]}((T'+R)-(R))$$

则 $E[n]$中共有 n^2 个元素和 $n^2T'=nT=O$。由定理3.2 可知，存在 $g\in\overline{F}(E)$，使得

$$D' = \mathrm{div}(g)$$

这里的g与T有关，下面用g_T表示。

关于f和g_T，容易证明

$$f \cdot [n] = g_T^n \tag{3.28}$$

这里，$(f \cdot [n])(X) = f(nX)$，$X \in E$。

由式(3.28)，对$E[n]$中另外的任意点$S \in E[n]$以及对E中任意点$X \in E$，有

$$\begin{aligned} g_T^n(X+S) &= f_T(nX+nS) \\ &= f_T(nX) \\ &= f_T \cdot [n](X) \\ &= g_T^n(X) \end{aligned}$$

即

$$\left(\frac{g_T(X+S)}{g_T(X)}\right)^n = 1$$

定义3.5 用μ_n表示$GF(q)$的代数闭域$\overline{F}_q$中n次单位根所构成的乘法群，令

$$e_n(S,T) = \frac{g_T(X+S)}{g_T(X)}$$

则对于$E[n]$中的任意两点$S,T \in E[n]$，可定义由$E[n] \times E[n]$到μ_n的映射Φ如下：

$$\begin{aligned} \Phi:\quad & E[n] \times E[n] \to \mu_n \\ & (S,T) \to e_n(S,T) = \frac{g_T(X+S)}{g_T(X)} \end{aligned}$$

这时，称映射Φ是一个Weil-配对。

Weil-配对实际上是一个函数，它有下列性质。

(1) 恒等性

对于任何$S \in E[n]$，有

$$e_n(S,S) = 1$$

(2) 双线性

对于任何$S_1,S_2,T\in E[n]$,有

$$e_n(S_1+S_2,T)=e_n(S_1,T)\cdot e_n(S_2,T)$$

$$e_n(S,T_1+T_2)=e_n(S,T_1)\cdot e_n(S,T_2)$$

(3) 交错性

对于任何$S,T\in E[n]$,有

$$e_n(S,T)=e_n(T,S)^{-1}$$

(4) 非退化性

对于任何$S\in E[n]$,有$e_n(S,O)=1$;反之,若对任何$S\in E[n]$,有$e_n(S,T)=1$,则$T=O$;

(5) 单位根

存在$T\in E[n]$,使得$e_n(S,T)$是一个n次单位根。

定理3.4　设$S\in E[n]$为一给定点,则存在另一点$T\in E[n]$,使$e_n(S,T)$是μ_n的本原元,特别地,对于阶为素数n的循环子群$\langle P\rangle\subset E[n]$,存在群元$R\in\langle P\rangle$,使得$e_n(P,R)\in\mu_n$是$\mu_n$的本原元。

由Weil-配对的构造过程及其性质可以证明定理3.4。

定理3.4给出了素数阶循环子群$\langle P\rangle$与μ_n之间的同构关系:

$$\Phi:\ \langle P\rangle\rightarrow\mu_n\subset\overline{F}_q$$

$$kP\rightarrow e_n(kP,R)=e_n(P,R)^k$$

对定义特征为p的有限域$GF(q)$上的椭圆曲线E的素数阶循环子群$\langle P\rangle$上的椭圆曲线离散对数问题,由定理3.4,可将椭圆曲线离散对数方程式(3.27)演化为域$\overline{F}_q$上的由n次单位根所组成的循环子群μ_n上的一个离散对数问题。具体方法如下。

对点$P\in E(GF(q))$,由定理3.4可知,存在$R\in E[n]$,使得$e_n(P,R)\in\mu_n$是μ_n的本原元。令$\alpha=e_n(P,R)$,$\beta=e_n(Q,R)$,则α为μ_n的本原元。由Weil-配对的性质可知

$$\begin{aligned}\beta &= e_n(Q,R)\\ &= e_n(lP,R)\\ &= e_n(P,R)^l\\ &= \alpha^l\end{aligned}$$

即

$$\beta = \alpha^l \tag{3.29}$$

式中，$\alpha,\beta\in\mu_n\subset\overline{F}_q$。

至此，已将基于$E(GF(q))$的椭圆曲线离散对数式(3.27)演化成了域$\overline{F}_q$上的普通离散对数式(3.29)，该演化过程可以在多项式时间内完成。

但是，对域$\overline{F}_q$上的普通离散对数式(3.29)的求解，有时并不比直接求解$E(GF(q))$上的椭圆曲线离散对数式(3.27)容易。为此，有下面的定理。

定理3.5　对定义在$GF(q)$上的椭圆曲线E，若存在k，使得$E[n]\subset E(GF(q^k))$，则在Weil-配对中，对于任何$S,T\in E[n]$，有$e_n(S,T)\in E(GF(q^k))$。

由定理3.5可知，当$E[n]\subset E(GF(q^k))$时，对方程式(3.29)中的α,β，有$\alpha,\beta\in EGF(q^k)$。这时，域$\overline{F}_q$上的普通离散对数式(3.29)实际上只是$GF(q^k)$上的一个离散对数问题，即对离散对数式(3.29)的求解实际上只需在$GF(q)$的某个扩域上进行就可以了。

另一方面，为了使方程式(3.29)变得更容易求解，应当使$E[n]\subset E(GF(q^k))$成立的k尽可能小。

为了进一步讨论MOV方法的适用范围，首先提出有限域上椭圆曲线超奇异性的概念。

定义3.6　对于定义特征为p的有限域$GF(q)$上的椭圆曲线E，当$E[p]=\{O\}$时，称E是超奇异的(Supersingular)；否则，称E是非超奇异的(Non-Supersingular)。

因此，对于定义于一个固定有限域上的全部椭圆曲线，总可以将它们分为两类。其中一类是所谓的超奇异型(Supersingular)椭圆曲线，另一类是所谓的非超奇异型(Non-Supersingular)椭圆曲线。

由定理3.3可知，超奇异型的各类椭圆曲线对任何不同于基域$GF(q)$的特征p的整数n，使式(3.30)成立的k必须满足$k\leqslant 6$。

$$E[n]\subset E(GF(q^k)) \tag{3.30}$$

通过MOV方法可将超奇异型椭圆曲线离散对数式(3.27)演化成至多只是有限域$GF(q^k)$上的一个普通离散对数式(3.29)，并可以在亚指数时间内通过Index算法得到有效地求解。

由此可知，超奇异型椭圆曲线在密码学上是不安全的。判定一条给定椭圆曲线的超奇异性成为椭圆曲线密码学中的一个基本问题。

那么如何判定椭圆曲线的超奇异性呢?首先引入Frobenius自同态的概念，然后再导出相关定理，来判定给定椭圆曲线的超奇异性。

定义3.7　对于定义特征为p的有限域$GF(q)$上的椭圆曲线E中的任意元素$P=(x,y)$作映射

$$\varphi:(x,y)\rightarrow(x^q,y^q),O\rightarrow O$$

或在射影坐标下，对E中的任意元素$P=(X,Y,Z)$作映射

$$\varphi:(X,Y,Z)\rightarrow(X^q,Y^q,Z^q)$$

对方程式(2.7)中的a_i，由$a_i^q=a_i$可知，φ是E的一个态射，从而是E的一个自同态，这时，称这一自同态是椭圆曲线E的Frobenius自同态，并称整数$t=q+1-\#E(GF(q))$为E的Frobenius自同态的迹。

对于有限域$GF(q)$和Frobenius自同态的迹t，不等式$|t|<2\sqrt{q}$恒成立，这一性质也称为Hasses定理。

由定义 3.7 可得下面的判定定理。

定理3.6 设E是定义特征为p的有限域$GF(q)$上椭圆曲线，当且仅当下列条件之一成立时，E是超奇异的。

① 当p为2或3时，$j(E)=0$；

② 当$p\geqslant 5$时，Frobenius 自同态的迹$t=0$；

③ 有限域$GF(q)$的特征p整除Frobenius 自同态的迹t。

依据定理 3.6 来判断给定椭圆曲线的超奇异性，避免因选用超奇异型的椭圆曲线而带来安全隐患。

另一方面，Menezes 指出非超奇异型的椭圆曲线要使式(3.30)成立，k将会变得非常大，使得对式(3.29)的求解难度大大高于直接对$E(GF(q))$上的椭圆曲线离散对数式(3.27)的求解难度，导致从式(3.27)到式(3.29)的 MOV 演化没有意义。所以，MOV 方法对基于非超奇异型椭圆曲线上的椭圆曲线离散对数问题无效。

2. SSAS 算法

1998 年，Smart、Semaev、Satoh 和 Araki 各自独立地提出了对素域上的一类“畸形”椭圆曲线上的离散对数问题的求解攻击算法，所以该算法被合称为 Smart-Satoh-Araki-Semaev 算法，简称 SSAS 算法。该算法针对素域$GF(p)$上的“畸形”(Anomalous)椭圆曲线，通过构造$E(GF(p))$到$GF(p)$的加法群上的一个同构映射，使得这类椭圆曲线离散对数问题的求解可以在多项式时间内完成。其中，Smart、Satoh 和 Araki 采用了代数数论的方法，而 Semaev 则采用代数几何的方法来完成这一映射的构造。

“畸形”椭圆曲线的定义如下。

定义3.8 对定义在有限域$GF(q)$上的椭圆曲线E，当E的 Frobenius 自同态的迹$t=1$时，称椭圆曲线E是“畸形”的或者说是

“非正规”的。

由定义3.7和定义3.8可知，迹$t=q+1-\#E(GF(q))=1$，故对于$GF(q)$上的“畸形”椭圆曲线E，有$\#E(GF(q))=q$。

定义3.9　对于定义在有限域$GF(q)$上的“畸形”椭圆曲线E及其在$GF(q)$的某个扩域$GF(q^n)$上的有理点所构成的子群$E(GF(q^n))$，曲线$E(GF(q^n))$的Frobenius自同态的迹$t_n=q^n+1-\#E(GF(q^n))$。这时，若$t_n=1$，则称$E(GF(q^n))$是一条“纯畸形”(Pure Anomalous)的椭圆曲线；反之，称$E(GF(q^n))$是一条“半纯畸形”的(Sub-Pure Anomalous)椭圆曲线。

设p是一素数，椭圆曲线E定义在有限域$GF(p)$上，且$\#E(GF(p))=p$，则定义在有限域$GF(p)$上的离散对数问题可描述如下。

对于给定的$P,Q\in E(GF(p))$，求正整数$m\in(0,p)$，使$Q=mP$。

1999年，Smart等人指出，若存在一种同构映射，能将有限域$GF(p)$上的椭圆曲线离散对数问题映射为有限域$GF(p)^*$上的离散对数问题$a^m\equiv\beta \bmod p$，则可通过求解$GF(p)^*$上的离散对数问题$a^m\equiv\beta \bmod p$，最终实现有限域$GF(p)$中的椭圆曲线离散对数问题$Q=mP$的求解。这种映射被称为“对数映射”。显然，对定义在有限域$GF(p)$上的椭圆曲线$E(GF(p))$，这种对数映射是不存在的。但对于有理数域上的椭圆曲线，标准对数映射是存在的。为此，Smart等人构造了针对有限域$GF(p)$中的椭圆曲线离散对数问题$Q=mP$的求解方法。

首先，采用Silverman提出的方法，将$P,Q\in E(GF(p))$定义为有理数域Q_p上的椭圆曲线$\tilde{E}$中的两点$\tilde{P},\tilde{Q}$。椭圆曲线$\tilde{E}$经过模p约化后所得的曲线就是E；而点$\tilde{P},\tilde{Q}$经过模p约化后所得的点

就是P,Q。

这时，有 $\tilde{Q}-m\ \tilde{P}=\tilde{R}\in\tilde{E}_1(Q_p)$，以及$\dfrac{\tilde{E}_0(Q_p)}{\tilde{E}_1(Q_p)}\cong\tilde{E}(F_p)$和$\dfrac{\tilde{E}_1(Q_p)}{\tilde{E}_2(Q_p)}\cong F_p^+$。对“畸形”椭圆曲线$E$，由于$E(GF(p))$和$GF(p)^*$中都只有$p$个点以及$\tilde{E}(F_p)=E(F_p)$，所以，有

$$p\tilde{Q}-m(p\tilde{P})=p\tilde{R}\in\tilde{E}_2(Q_p) \tag{3.31}$$

点$p\ \tilde{Q}$，$p\ \tilde{P}$和$p\ \tilde{R}$都属于$\tilde{E}_1(Q_p)$，故在$\tilde{E}_1(Q_p)$上对方程式(3.31)的两端同时使用p进制椭圆对数Ψ_p后，得

$$\Psi_p(p\tilde{Q})-m\Psi_p(p\tilde{P})=\Psi_p(p\tilde{R})\equiv 0 \bmod p^2 \tag{3.32}$$

故

$$m\equiv\frac{\Psi_p(p\tilde{Q})}{\Psi_p(p\tilde{P})} \bmod p$$

当点$(x,y)\in E_1(Q_p)$时，有

$$\Psi_p(x,y)\equiv\frac{-x}{y} \bmod p^2$$

由式(3.32)计算m的关键在于计算点$p\ \tilde{Q}$和$p\ \tilde{P}$。该计算能在$O(\lg p)$时间内完成。所以，由定义3.4可知，SSAS算法能够在多项式的时间内求出椭圆曲线离散对数m。

3. 其他求解算法

除了上面介绍的两种针对特殊类型椭圆曲线上的离散对数问题的求解攻击方法以外，近几年来，人们还提出了其他一些针对特殊类型椭圆曲线的攻击算法。这里将简单地介绍其中的几种攻击

算法。

1999年，Frey、Muller和Ruck提出了一种利用Tate-Pairing的求解椭圆曲线离散对数问题的攻击算法，该算法实际上是MOV方法的改进版本。在MOV方法中，由定理3.5，首先需要找到合适的k，以使$E[n]\subset E(GF(q^k))$条件成立。但在Frey等人提出的攻击算法中，仅仅要求$n|(q-1)$，亦即Frobenius自同态的迹$t=2$，而不再需要求出k值。所以，该攻击算法略大于MOV方法的适用范围，计算效率也高于MOV方法。

2000年，Gaudry、Hess和Smart提出了Weil下降理论，是基于有限扩域$GF(p^n)$中的椭圆曲线群上的离散对数问题的求解攻击方法，这里的$p=2^l$。该算法利用Weil下降理论，构造一个定义在有限域F上的Abelian簇$A(F)$，将定义在扩域$GF(p^n)$中的椭圆曲线$E(GF(p^n))$上的离散对数问题归约到子域代数簇$J_{ac}(c)(GF(p))$上的离散对数问题，然后用Index算法求解$Jac(c)(GF(p))$上的离散对数问题即可。

Weil下降攻击算法的实现依赖于下面三个问题的求解。

① 找出Abelian簇A上具有小亏格的代数曲线；

② 找出簇A上的点所对应的除子；

③ 找出解决广义除子类群离散对数问题的Index算法。

Weil下降算法能够将椭圆曲线离散对数问题归约成超椭圆曲线离散对数问题，而对于高亏格的超椭圆曲线离散对数问题的求解则是存在亚指数时间攻击的。2001年，加拿大的Jacobson和Menezes利用Weil下降算法和超椭圆曲线离散对数Index指标求解算法，分析和计算了特征为2的有限域上的椭圆曲线C_{62}，C_{93}，C_{124}和C_{155}上的离散对数问题。得出结论：若用1 000台主频为1 GHz的Pentium III计算机组成的计算网格来求解这些离散对数问题，则可以在一个月内解决C_{155}上的椭圆曲线离散对数问题。

不过，Weil 下降算法只适用于特征值为2 的复合域，并且当代数曲线的亏格较大时，求解上面三个问题是不可行的；而亏格较小（亏格 $g<4$）时，对这些代数曲线上的超椭圆曲线离散对数问题的攻击，目前最好的算法还是分布式Pollard 方法。为避免这一攻击，应该避免选用特征为 2 的复合域，即所选椭圆曲线的基域 $GF(2^p)$ 中的 p 应该为素数。

3.5 安全椭圆曲线

由于椭圆曲线密码体系建立在有限域上的椭圆曲线离散对数问题的基础上，故从安全性角度来看，该问题越难越好。在同等规模的椭圆曲线离散对数问题中，如果基于某条椭圆曲线的离散对数问题的求解困难越大，则该椭圆曲线越安全。

一般地，对给定的椭圆曲线 E，若求解其上的离散对数问题需要指数时间，则称该椭圆曲线是安全的。从应用角度看，安全椭圆曲线在构造椭圆曲线密码体系时是可以选用的。

现设定义于有限域 $GF(q)$ 上的椭圆曲线 E，其中 $q=p^n$，p 是一素数。椭圆曲线 E 的有理子群 $E(GF(q))$ 的阶用 $\#E(GF(q))$ 来表示，设 l 是 $\#E(GF(q))$ 的最大素因子，对于 $E(GF(q))$ 上的椭圆曲线离散对数问题 $Q=mP$，由前面两节可知，目前最好的求解方法有以下 4 种。

1. Pohlig-Hellman 算法

Pohlig-Hellman 算法适用于任何有限 Abel 群上的离散对数问题，它采用中国古代剩余定理，将 N 阶有限群 G 上的离散对数问题转化为素因子的循环子群上的若干离散对数问题。当生成元的

阶的因子全部是小素数因子时，这种算法非常有效。

2. Pollard-ρ 算法

Pollard-ρ 算法对任何基于有限 Abel 群上的离散对数问题都有效。在通过Pohlig-Hellman 算法约简后，可以在 l 素数阶子群上求解新的椭圆曲线离散对数问题。针对椭圆曲线的特殊性，Pollard-ρ 算法的运行时间复杂度为 $O\left(\sqrt{\frac{\pi l}{4}}\right)$，空间复杂度可忽略不计。并行化处理后的分布式Pollard-ρ 算法的运行时间复杂度可降至 $O\left(\sqrt{\frac{\pi l}{2M}}\right)$。但该算法本质上仍是指数时间算法。

3. MOV 方法

对于使 $E[n]\subset E(GF(q^k))$ 成立的最小 k，可通过MOV 方法在多项式时间内将椭圆曲线离散对数问题约化为有限域 $GF(q^k)$ 上的离散对数问题。当 k 不很大时，对约化后的DLP 问题，利用Index 算法，可在亚指数时间内得到解决，进而可得到原 ECDLP 问题的解。

但MOV 方法仅对超奇异型椭圆曲线以及 Frobenius 自同态的迹 $t=2$ 的椭圆曲线有效，对其他类型的椭圆曲线无效。

4. SSAS 方法

对“畸形”椭圆曲线，该算法可在多项式时间内求解其上的椭圆曲线离散对数问题。此时，Frobenius 自同态的迹 $t=1$ 或 $\#E(GF(q))=q$。因此，当椭圆曲线 E 的Frobenius 自同态的迹 $t=0,1,2$ 以及 $p\mid t$ 时，上述的攻击方法可以在低于指数的时间内求解 ECDLP 问题，这样的椭圆曲线 E 是不能用来构造密码体系的。就

目前已知的针对椭圆曲线离散对数的攻击方法而言，在选取安全的椭圆曲线 $E(GF(q))$ 时，应该遵循下列两条选取准则。

① 椭圆曲线 $E(GF(q))$ 的Frobenius 自同态的迹 $t \neq 0,1,2$ 且 p 不能整除 t；

② 该椭圆曲线群的阶 $\#E(GF(q))$ 必须包含一个大素数因子，以避免 Pohlig-Hellman 算法的攻击，加大 Pollard-ρ 算法攻击的困难性。

依据上面的安全椭圆曲线选取准则，在实际判定所选取的椭圆曲线 $E(GF(q))$ 的安全性时，首先要计算该椭圆曲线曲线群的阶 $\#E(GF(q))$ 和Frobenius 自同态的迹 $t=q+1-\#E(GF(q))$，然后判断 t 是否满足安全准则①。当满足该准则时，再对 $\#E(GF(q))$ 进行分解，判断其是否含有一个大于 l 的大素数因子，若仅有一个大于 l 的大素数因子，则接受该椭圆曲线作为安全椭圆曲线。

这里的阈值 l 的确定是比较复杂的。为了保证系统的整体工作性能，应该在满足一定的安全性要求的前提下，使 l 尽可能小。

2000 年，Lenstra 和 Verheul 对如何确定 l 的值做了比较详细的讨论并指出：决定最低安全性要求的因素应包括以下几个方面。

① 被保护信息的重要程度；

② 现有计算机的计算能力；

③ 对未来计算能力提高的预测；

④ 对新的求解攻击算法出现情况的估计。

由此可知，依据上述安全椭圆曲线的选取准则，可以细分有限域上的安全椭圆曲线的类型如下。

第一类：有限域的特征为素数 p 的椭圆曲线。这类曲线又可细分为如下几种。

① $q=p$，$GF(q)=GF(p)$，这时，椭圆曲线 E 定义在素域

$GF(p)$上，其Weierstrass方程如式(2.14)。这里的p是一个足够大的素数，以使椭圆曲线群E的阶$\#E(GF(q))$包含一个大素数因子，最好$\#E(GF(q))$就是素数，这时，称该曲线是“理想”的安全椭圆曲线。

② $q=p$，$GF(q)=GF(p)$，p是一个较小的素数(如$p=3,5,7,11$等)，对定义在素域$GF(p)$上的椭圆曲线E，其Weierstrass方程如式(2.14)。这时，对某个素数n，当E在$GF(p)$的扩域$GF(p^n)$上的椭圆曲线有理子群$E(GF(p^n))$，当$\#E(GF(p^n))$中包含有大的素数因子时，$E(GF(p^n))$可以用来构造椭圆曲线密码体系。从提高系统的运算性能着想，一般要求p的大小接近于使用该体制的CPU所能处理的最大字长的一个素数。例如，当使用该密码体系的CPU的字长是8位时，选取接近256的素数，如257。

第二类：有限域的特征为2的椭圆曲线。这类曲线也可细分为如下几种。

① $q=2^n$时，椭圆曲线E定义在二元有限域$GF(q)$上，其Weierstrass方程如式(2.15)。为对抗Weil下降算法，这里的n是素数，并且2^n应足够大，以使$\#E(GF(2^n))$中含有一个大素数因子。

② $q=2$时，椭圆曲线E定义在有限域$GF(2)$上，这时的非超奇异型椭圆曲线只有以下两个。

$$\begin{aligned} E_1&: y^2+xy=x^3+x^2+1 \\ E_2&: y^2+xy=x^3+1 \end{aligned} \tag{3.33}$$

对某个素数n，E_i在$GF(2)$的扩域$GF(2^n)$上的有理子群为$E_i(GF(2^n))$。当$\#E_i(GF(2^n))$中包含有大素数因子时，可用$E(GF(2^n))$来构造密码体系，一般也称这种曲线为Koblitz曲线。

③ $q=2^r$时，椭圆曲线E定义在二元有限域$GF(q)$上，其Weierstrass方程如式(2.15)。为提高系统的运行性能，一般取r与

使用该体系的CPU 的字长一样。例如，对32 位CPU，取$r=32$。再对某个素数n，考虑E 在$GF(q)$的扩域$GF(q^n)$上的有理子群$E(GF(q^n))$。当$\#E(GF(q^n))$中包含有大的素数因子时，$E(GF(q^n))$可以用来构造椭圆曲线密码体系。

④ $q=2^n$时，$n=ed$，e，d 一般是素数，E 定义在二元有限域$GF(q)=GF(2^n)$上，当$\#E(GF(q))$中包含有大的素数因子时，$E(GF(q))$可用来构造密码体系。

在上述的各种椭圆曲线中，第一类的第①种曲线和第二类的第①种曲线是两种最典型、最基本的安全椭圆曲线。选用这两种曲线来构造密码体系有下列一些优点。

● 可灵活选取基域$GF(p)$和$GF(2^n)$中p 或2^n 的大小。

对素数p，可取p 为Fermat 素数、Mersenne 素数等，如取$p=2^{196}-c$，c 为很小的整数；而对$GF(2^n)$中的2^n，可取n 为接近2 的m 次幂的素数。这样，可以大大提高系统的运行速度。

● 曲线选取的空间足够大。

对于选定的基域$GF(p)$和$GF(2^n)$，由于p 或2^n 非常大，所以在有限域$GF(p)$和$GF(2^n)$上有足够多的椭圆曲线，使得安全椭圆曲线的选取空间足够大，能够满足各种场合应用的需要。

● 安全性最好。

目前一般认为，基于随机选取的安全椭圆曲线所构造的椭圆曲线密码体系在各类椭圆曲线中具有最好的安全性。

但另一方面，选用这两种类型的椭圆曲线也有两个明显的缺点。

● 曲线选取困难。

在选用这两种类型的椭圆曲线时，必须首先计算椭圆曲线群的阶$\#E(GF(p))$或$\#E(GF(2^n))$。虽然在目前用于计算椭圆曲线有限群的阶的SEA 数点算法（下一章讨论）已达到实用化的程

度，但该算法仍然非常复杂。而且，当利用SEA算法求出阶$\#E(GF(p))$或$\#E(GF(2^n))$后，还需要对所求得的$\#E$进行分解，以确定其中是否含有一个大素数因子。

实验数据表明，在随机选取的这两种类型曲线中，大约每1 000条曲线中仅有两条曲线能满足安全性要求。这说明从随机产生的这两种类型的椭圆曲线中找出安全椭圆曲线的工作是非常复杂的，采用这两种曲线构造密码安全体系时，曲线选取是一个严重的问题。

● 运行速度较慢。

这两种曲线实现起来的运行速度普遍比较慢的原因在于：一是描述该椭圆曲线的Weierstrass方程一般有很大的系数，因而在一定程度上降低了其上的群运算速度；二是对这两种类型的椭圆曲线上的数乘运算，除了用在第6.4节中所研究的方法以外，目前还没有更好的算法。

除了上述两种类型曲线以外，其他各种曲线都有一个共同特点，即定义该曲线的Weierstrass方程所使用的有限域$GF(q)$都是使用该曲线时所涉及的有限域$GF(q^n)$的一个子域，故统称这些曲线为子域曲线，也称为特殊曲线。对于子域曲线，一般要求域$GF(q^n)$中的n是一个素数，以增强安全性。

选用子域曲线来构造密码体系有如下两个优点。

● 椭圆曲线有限群$E(GF(q^n))$的阶容易计算，能够比较容易地找到合适的n，使得$\#E(GF(q^n))$中包含一个大素数因子。

● 这种椭圆曲线上的数乘群运算容易进行。例如，对Koblitz曲线使用第6.4节中的二进制算法做数乘运算时，一般需要$\frac{4}{3}n$次点加或倍点运算；而使用Frobenius展式时，仅需$\frac{1}{3}n$次点加或倍点运算。

但是，选用这种子域曲线也有如下两个缺点。

● 这种曲线太稀少。如对Koblitz 曲线E_1，仅当n取3，5，7，11，17，23，101，109，163，283，311，331，359 等素数时，$\#E_1(GF(2^n))$是2 与某一素数的乘积；而对于Koblitz 曲线E_2，仅当n取5，7，13，19，23，41，97，103，107，131，233，239，277，283，349，409 等素数时，$\#E_1(GF(2^n))$是4 与某一素数的乘积。

● 有安全性顾虑。安全性顾虑主要体现在以下两个方面：一方面，人们担心这种曲线所具有的特殊结构很有可能成为攻击者的突破口，从而使得基于该曲线的椭圆曲线密码体系具有较少的安全性；另一方面，由于这种曲线的数量稀少，故一些特殊机构可能对具体的每一条曲线展开专门攻击，而且同一条曲线的大量重复使用也会为攻击者提供大批可利用的数据。

因此，在本书中，作者倾向于选用由完全随机产生的理想安全椭圆曲线来构造椭圆曲线密码体系。

第4章　椭圆曲线有限群阶的计算

安全椭圆曲线参数选取的核心问题是寻找合适的具有素数阶或拟素数阶的椭圆曲线，它可以归结为对给定椭圆曲线有限群的阶的计算。SEA 算法是目前唯一在理论上比较成熟的、能够较好地求解椭圆曲线有限群上数点问题的有效算法，但其实现却是一个相当复杂的问题。

本章首先介绍Schoof 算法和SEA 算法的算法框架，并对它们进行分析。在此基础上，研究SEA 算法实现的具体细节问题。另外还将介绍 Morain 对 Elkies 算法改进之后所得到的同种圈算法、当 l 为 Atkin 素数时的 Atkin 算法、利用大步小步法和中国古代剩余定理及Hasse 定理完成SEA 算法的最后阶段并求解出给定椭圆曲线有限群的阶的方法。

理论上，椭圆曲线密码体系的安全性可以归结为对椭圆曲线有限群参数的选择。由前文可知，参数选取的核心问题在于寻找合适的具有素数阶或拟素数阶的椭圆曲线。目前，一般有两种方法可用于解决这一问题。

① 通过构造方法来得到指定阶的椭圆曲线，其中最著名的就是复乘(Complex Multiplication，CM)方法。

② 随机选取定义在给定有限域上的椭圆曲线，计算该曲线的阶，直到其满足第 3.5 节所介绍的安全椭圆曲线的选取准则为止。

第①种方法相对比较简单，生成速度也比较快，能够应付现有

的几种攻击。但由这类方法所构造的椭圆曲线一般具有一些附带的结构特征，如复乘方法所构造的曲线就具有复乘特征，且与虚二次域的某个阶有内在的联系，这些特征从安全角度上来看有潜在的威胁，建立在其上的密码体系也具有潜在的不安全性隐患。而由第②种方法所得到的椭圆曲线完全是随机选取的，不存在上述的各种潜在威胁，因而是比较理想的，目前被理论界公认为是安全性最好的。

但第②种随机选取椭圆曲线的方法则完全依赖于椭圆曲线有限群的阶的计算，这一问题通常被称为椭圆曲线的数点问题(Elliptic Curve Point Counting Problem，ECPCP)。与椭圆曲线离散对数问题一样，它也是椭圆曲线密码体系设计中的基本问题之一。

在这一方面，R. Schoof 做出了具有开创意义的工作。1985年，他提出了著名的Schoof 算法，该算法可在$O(\lg^8 q)$次基本域元素计算的时间内确定椭圆曲线有限群 $E(GF(q))$ 的阶 $\#E(GF(q))$，但对实际应用中的q，该算法实际上不可行。之后，Elkies 和 Atkin 对 Schoof 算法做了重大的改进，使算法的时间复杂度降至 $O(\lg^6 q)$，对实际应用中的 q，可以较快地求解 $\#E(GF(q))$。因而，目前一般将 Schoof 算法连同 Elkies 和 Atkin 的改进结合起来，即SEA 数点算法。1999 年，Satoh 提出了一种全新的数点算法。2000 年 5 月，Harly 和 Gaudry 等人利用这一方法在 54 小时内求出了定义在有限域 $GF(2^{3001})$ 上的随机椭圆曲线有限群阶的计算。同年10 月，他们又宣布完成了有限域 $GF(2^{8009})$ 上的ECPCP 的求解。但考虑到Satoh 算法目前只适用于解决基域为 $GF(2^m)$ 上的ECPCP 的求解，所以本书暂不做讨论。

1995 年，Schoof 对素数有限域下的SEA 算法进行了初步总结；而对一般的复合域 $GF(p^n)$ 上的数点问题，Couvergnes 在 1995

年提出了形式群方法，1996年提出了 p 次挠子群方法；同时，Lercier也提出了针对二次有限域$GF(2^n)$上的数点问题的方法等。这些工作进一步完善了Elkies的改进，使得SEA算法能够应用于复合域上的数点问题的求解。之后，Muller、Morain、Dewaghe等学者也对SEA算法进行了大量的优化，做了许多卓有成效的工作。至2002年底，已经成功地完成了对定义在有限域$GF(2^{130020})$上的随机椭圆曲线有限群阶的计算。这说明，继Schoof的开创性工作之后，经过十几年的不懈努力，目前，SEA算法已经比较成熟，能够较好地解决随机椭圆曲线有限群上的数点问题，成为计算椭圆曲线有限群阶的有效算法。

尽管SEA算法在理论上是比较成熟的，但其实现仍是一个涉及深奥的数学理论、富有相当挑战性的复杂问题。目前，国内对SEA算法的实现水平与国际先进水平尚有相当的差距。

SEA算法是选择安全椭圆曲线、决定椭圆曲线密码体系安全性的核心算法之一，其实现对拥有我国自主知识产权的椭圆曲线密码体系和开发相关产品具有非常重要的意义。本章将在讨论Schoof和SEA算法框架的基础上，综合作者的研究实践工作，给出具体的实现方法。

4.1 Schoof算法

由定义3.7可知，对椭圆曲线有限群$E(GF(q))$有

$$\#E(GF(p)) = q + 1 - t$$

式中，t 为椭圆曲线 E 的Frobenius自同态的迹。

这样，计算椭圆曲线有限群$E(GF(q))$的阶$\#E(GF(q))$时只需求解 t 即可。而为了计算 t，可以对一些较小的素数$l=2,3,5,7,$

…分别计算 $t \bmod l$，当所计算出来的 $t \bmod l$ 足够多，即满足

$$\prod_{l\text{为素数}} l > 4\sqrt{q} \tag{4.1}$$

时，采用中国古代剩余定理CRT(定理3.1)即可唯一地确定 t。

这样，椭圆曲线有限群上的数点问题可归结为对 $t \bmod l$ 的计算。由素数定理可知，需要的素数个数为 $O(\lg q/\lg\lg q)$，其中最大的素数为 $O(\lg q)$。

现考虑素域上的椭圆曲线有限群 E，记 $E[l]=\{P\in E \mid lP=O\}$ 为曲线 E 的挠子群(l-torsion 点群)，则由曲线 E 的Frobenius 映射 $\varphi:(x,y)\to(x^p,y^p)$ 诱导出 Tate 模 $T_l(E)$ 上的线性映射，其特征方程满足 $\varphi^2-t\varphi+p=0$，式中 t 为椭圆曲线 E 的Frobenius 自同态的迹。为了计算 $t \bmod l$，设 $t_l\equiv t \bmod l$，故有

$$\varphi^2(P)-[t_l]\varphi(P)+[p](P)=O,\quad \forall P\in E[l] \tag{4.2}$$

记 $l-th$ 可除多项式为 $f_l(x)$，$\deg f_l=(l^2-1)/2$，且其根恰为 $E[l]\backslash\{O\}$ 中点的 x 坐标。因此，式(4.2)的计算是在环 $F_p[X,Y]/f_l(X)$ 中进行的。也就是说，通过上述步骤，将对 $t \bmod l$ 的计算转化为在环 $F_p[X,Y]/f_l(X)$ 中对多项式公因子的计算。

具体地说，在环 $F_p[X,Y]/f_l(X)$ 上对式(4.2)的计算即求解同余方程

$$(x^{p^2},y^{p^2})+[p_l](x,y)\equiv[t_l](x^p,y^p) \bmod (f_l(X)) \tag{4.3}$$

式中，$p_l\equiv p \bmod l$，$t_l\in\{0,1,\cdots,l-1\}$。

为了求解满足方程式(4.3)的 t_l，可利用穷举法将 $t_l=0,1,\cdots,l-1$ 逐一代入，验证方程式(4.3)是否成立，这样最多可在 l 次尝试后确定 $t_l\equiv t \bmod l$，进而可求解 t 和 $\#E(GF(q))$。实际实现时可利用椭圆曲线有限群运算中求逆元运算不费时间的特点，将尝试次数减半。

当 $l=2$ 时，情况更加简单。

由 p 为奇数、$\#E(GF(p))=p+1-t$ 可知，$t \bmod 2=\#E(GF(p)) \bmod 2$。显然，当且仅当椭圆曲线群 E 中存在一个二阶点 P 时，有 $\#E(GF(p)) \bmod 2=0$。此时二阶点 P 满足 $2P=O$。而由椭圆曲线群运算法则(式(3.2)、式(3.3))可知，$y=0$。即 $2P=O$ 等价于方程

$$x^3+ax+b=0 \tag{4.4}$$

在有限域 $GF(p)$ 上有解，亦即多项式 X^3+aX+b 与多项式 X^p-X 有次数不为零的公因式，即

$$t \bmod 2=\gcd(X^3+aX+b, X^p-X) \tag{4.5}$$

以上就是R. Schoof 于1985年提出的Schoof 算法，具体过程如下。

算法 4.1　Schoof 数点算法

输入：素域 $GF(p)$ 及椭圆曲线 E
输出：$\#E(GF(p))$

1. $M \leftarrow 2$；
2. $l \leftarrow 3$；
3. $S \leftarrow \{(t \bmod 2, 2)\}$[由方程式(4.5)计算]；
4. while $M<4\sqrt{p}$ do
   4.1　for $\tau=0,1,\cdots,\left[\frac{l-1}{2}\right]$ do
   (a) 检测 $\forall P \in E[l]$，$\varphi^2(P)+[p](P)=\pm[\tau]\varphi(P)$；
   (b) 若成立则终止循环；
   4.2　若 $\varphi^2(P)+[p](P)=[\tau]\varphi(P)$，则 $S \leftarrow S \cup \{(\tau, l)\}$，否则 $S \leftarrow S \cup \{(-\tau, l)\}$；
   4.3　$M \leftarrow M \times l$；
   4.4　取下一个素数 l；

5. 对余数集合S,由中国古代剩余定理3.1计算t;
6. 输出$\#E(GF(p))=p+1-t$。

由以上算法可知,在Schoof算法的整个工作过程中,关键运算在于求解$X^p \bmod f_l(X)$,而$\deg f_l(X)=\frac{l^2-1}{2}$,故当$l$比较大时,$X^p \bmod f_l(X)$的求解异常困难,整个算法工作将慢得难以忍受,故该算法实际上是不可行的。

4.2 SEA算法

在Schoof算法发表后的10余年中,Elkies和Atkin对Schoof算法进行了重大改进,最终形成了能有效计算有限域上椭圆曲线有理点个数的SEA算法。SEA算法中所用到的两类素数也因此被称为Elkies素数和Atkin素数。

定义4.1 对有限域$GF(p)$上的椭圆曲线E,其挠子群$E[l]=\{P\in E \mid lP=O\}$。当$E[l]$上的Frobenius映射在$f_l$中有特征值时,称$l$是Elkies素数;否则,称$l$是Atkin素数。这可由$\Phi_l(x,j(E))$的分裂形式来判断。

在对Schoof算法的改进中,希望通过对l次模多项式$\Phi_l(x,j(E))$的零点的考察来获得关于Frobenius映射φ的特征多项式$f_l(\lambda)=(\lambda^2-t\lambda+p) \bmod l$的根的性质,进而可确定$t \bmod l$的值或者可能值的集合。

定理4.1 设E定义在有限域$GF(p)$上且是非奇异的,$j(E)\neq 0$,对E的l次模多项式$\Phi_l(x,j(E))$,设$\Phi_l(x,j(E))$有素因子分解式:

$$\Phi_l(x, j(E)) = \prod_{i=1}^{s} h_i(x) \tag{4.6}$$

对于$h_i(x)$的次数以及$f_l(x)$的根，有如下几种类型。

① $s=2$，且$\deg h_1(x)=1, \deg h_2(x)=l$。为方便起见，可简记为$(1,l)$型(以下类似)。这时，特征多项式$f_l(x)=(x-\alpha)^2$，特征方程$f_l(x)=0$有两重根，其判别式$\Delta=t^2-4p\equiv 0 \bmod l$，故$4p$是有限域$GF(l)$上的一个平方剩余，即$t=\pm 2\sqrt{p} \bmod l$。

② $(1,1,\cdots,1)$型。即$\deg h_1(x)=\deg h_2(x)=\cdots=\deg h_{l+1}(x)=1$。这时与类型①一样，有$f_l(x)=(x-\alpha)^2$，$t=\pm 2\sqrt{p} \bmod l$。

③ $(1,1,r,r,\cdots,r)$型。这时，特征多项式$f_l(x)=(x-\alpha)\cdot(x-\beta)$，特征方程$f_l(x)=0$有两个不等的有理根$\lambda_1$和$\lambda_2$，故$t^2-4p$是有限域$GF(l)$中的一个平方剩余，且$r \mid l-1$，$\varphi$作为$E[l]$上的线性变换时，其矩阵相似于$\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}$。此时，有$t\equiv\lambda+\frac{p}{\lambda} \pmod l$，其中$\lambda$可由Elkies方法得到。显然，类型①和类型②均可被并入这一类型，此时$\lambda=\sqrt{p}$。

④ $(r,r,\cdots,r)$型，其中$r>1$。这时，特征多项式$f_l(x)$不可约，特征方程$f_l(x)=0$无有理根，t^2-4p在有限域$GF(l)$是一个非平方剩余，$r \mid l+1$。此时，可用Atkin方法求解。

为了完成l次模多项式$\Phi_l(x,j(E))$的分解，即求解$\Phi_l(x,j(E))$在$GF(p)$中的有理根，可通过计算$\gcd(x^p-x, \Phi_l(x,j(E)))$得到。当$\gcd(x^p-x, \Phi_l(x,j(E)))\neq 1$时，说明$\Phi_l(x,j(E))$在$GF(p)$中有零点，$l$为Elkies素数，可用Elkies方法求解；反之，若$\gcd(x^p-x, \Phi_l(x,j(E)))=1$，则$l$为Atkin素数，用Atkin方法求解$t \bmod l$的一个可能值的集合。

综上所述，SEA算法框架如下。

算法 4.2 SEA 数点算法

输入：素域 $GF(p)$ 及椭圆曲线 E

输出：$\#E(GF(p))$

1. $M \leftarrow 2$；
2. $l \leftarrow 3$；
3. $E \leftarrow \{(t \bmod 2, 2)\}$[由方程式(4.5)计算]；
4. $A \leftarrow \{\}$；
5. while $M < 4\sqrt{p}$ do
   - 5.1 计算 $\gcd(x^p - x, \Phi_l(x, j(E)))$；
   - 5.2 若 $\gcd(x^p - x, \Phi_l(x, j(E))) \neq 1$，则由 Elkies 改进：
     - (a) 求解 $\gcd(x^p - x, \Phi_l(x, j(E)))$ 的一个根；
     - (b) 计算 $t \equiv \left(\lambda + \frac{p}{\lambda}\right) \bmod l$；
     - (c) $E \leftarrow E \cup \{(t, l)\}$；
   - 5.3 若 $\gcd(x^p - x, \Phi_l(x, j(E))) = 1$，则由 Atkin 改进：
     - (a) 求解 $t \bmod l$ 的一个可能值的集合 T；
     - (b) $A \leftarrow A \cup \{(T, l)\}$；
   - 5.4 $M \leftarrow M \times l$；
   - 5.5 取下一个素数 l；
6. 对余数集合 E、A，由中国古代剩余定理 3.1 和大步小步算法 BSGS 计算 t；
7. 输出 $\#E(GF(p)) = p + 1 - t$。

以上算法仅仅是SEA 算法的框架，要完全实现SEA 算法，还有许多工作要做。在随后的几节中，将完善这一框架，分别给出各子算法的具体实现以及SEA 算法的改进。

4.3 模多项式及其实现

模多项式在求解椭圆曲线有限群的数点问题中占有非常重要的地位。从理论上说，模多项式是一个完全独立于SEA数点算法的概念，可以从SEA算法中分离出来单独研究。针对不同的素数l，只要求出模多项式$\Phi_l(x,y)$之后，对任何素数p以及定义在素域$GF(p)$上的任何椭圆曲线E，当使用SEA算法求解$\#E(GF(p))$时，所需要的模多项式都是相同的$\Phi_l(x,y)$。

本节将根据SEA算法的需要，在研究模多项式理论的基础上，指出求解模多项式的困难性，给出解决它的方法和具体的实现算法。

现用C表示复数域，$L=[1,\tau](\tau\in C,\mathrm{Im}(\tau)>0)$为$C$上的一个格，由代数曲线理论可知，一条定义在$C$上的椭圆曲线$E$一定与$C/L$同构。此时，曲线$E$的$j$不变量是0权的模函数$j(\tau)$。假设$l$是一素数，则$E$的所有$l$次同种曲线所对应的$j$不变量分别是$j\left(\frac{\tau+n}{l}\right)$和$j(l\tau)$，式中$0\leqslant n<l$。

若定义模多项式

$$\Phi_l(x,j(\tau))=(x-j(\tau))\prod_{n=0}^{l-1}\left(x-j\left(\frac{\tau+n}{l}\right)\right) \tag{4.7}$$

则易知$\Phi_l(x,j(\tau))$实际上是关于x和j的一个整次数多项式，且x和j对称。用y替代$\Phi_l(x,j(\tau))$中的j，即可得到$Z[x,y]$上关于x和y对称的多项式$\Phi_l(x,y)$，称$\Phi_l(x,y)$为C上的l次模多项式。

一般地，l次模多项式$\Phi_l(x,y)$是一个由l唯一确定的整系数多项式，其形式一般如下。

$$\Phi_l(x,y) = x^{l+1} + y^{l+1} + \sum_{m=0}^{l}\sum_{n=0}^{l} a_{mn}x^m y^m \qquad (4.8)$$

式中，$a_{mn}\in Z$ 且 $a_{mn}=a_{nm}$。

就SEA 算法中所涉及的素数 l 而言，实现 $\Phi_l(x,y)$ 的求解算法是一个复杂且困难的问题。Elkies 描述了求解 $\Phi_l(x,y)$ 的方法，并指出其时间复杂度为 $O(l^4)$。虽然，l 次模多项式 $\Phi_l(x,y)$ 的次数不超过 $l+1$，但其中的系数 $a_{mn}\in Z$ 随着 l 的增加而迅速增长。例如，当 $l=2,3,5$ 时，有

$$\begin{aligned}\Phi_2(x,y)=&x^3+y^3-x^2y^2+1488(xy^2+x^2y)-162000(x^2+y^2)\\&+40773375xy+8748000000(x+y)\\&-157464000000000\end{aligned}$$

$$\begin{aligned}\Phi_3(x,y)=&x^4+y^4-x^3y^3+2332(x^2y^3+x^3y^2)\\&-1069956(xy^3+x^3y)+3686400(x^3+y^3)\\&+2587918086x^2y^2+89002220976000(x^2y+xy^2)\\&+770845966336000000xy\\&+1855425871872000000000(x+y)\end{aligned}$$

$$\begin{aligned}\Phi_5(x,y)=&x^6+y^6-x^5y^5+3720(x^5y^4+x^4y^5)\\&-4550940(x^5y^3+x^3y^5)+2028551200(x^5y^2+x^2y^5)\\&-246683410950(x^5y+xy^5)+1963211489280(x^5+y^5)\\&+1665999364600x^4y^4+1078789281853368000(x^4y^3+x^3y^4)\\&+\cdots\\&+532743308034244254504201602733565091512320000(x+y)\\&+1413599471547213586977534746910713627510046720000\end{aligned}$$

一般地，若用 $d(\Phi_l)$ 表示 $\Phi_l(x,y)$ 中最大系数的绝对值的自然对数时，Blake 指出，$d(\Phi_l)$ 与 l 有下列估计。

$$d(\Phi_l) = 6(l+1)\left(1-\frac{2}{l}\right)\lg l + O(1)$$

由此可知，模多项式$\Phi_l(x,y)$的系数a_{mn}随着l的增加呈指数级增长。到1993年年底，只能求出41次模多项式$\Phi_l(x,y)$。而$l=41$对SEA算法而言是远远不够的，这样，模多项式$\Phi_l(x,y)$中巨大的系数a_{mn}已经成为实现SEA算法的重要障碍。

为解决这一问题，考虑到SEA算法只用到了模多项式$\Phi_l(x,y)$的分解模式，所以，希望构造一种新的模多项式，使它与原始的模多项式具有相同的分解模式，同时又具有相对较小的系数。在这一方面，Elkies、Muller和Couveignes分别提出了有关新型模多项式$\Phi_l(x,y)$的构造方案，但计算模多项式的工作量仍然相当大。

下面就基于Muller的工作给出具体的构造方案和实现算法。

对给定素数l，设$s=\dfrac{12}{\gcd(l-1,12)}$，$v=\dfrac{l-1}{12}s$，并定义

$$f(\tau)=\left(\frac{\eta(\tau)}{\eta(l\tau)}\right)^{2s}$$

式中，Dedekindη-函数$\eta(\tau)=q^{\frac{1}{24}}\prod_{n=1}^{\infty}(1-q^n)$。

这样，有下面的定理。

定理4.2　对模函数$j(l\tau)$和$f(\tau)$，存在整系数$a_{rk}\in Z$，使得下式成立。

$$\sum_{r=0}^{l+1}\sum_{k=0}^{v}a_{rk}j(l\tau)^k f(\tau)^r=0$$

现定义多项式

$$G_l(x,y)=\sum_{r=0}^{l+1}\sum_{k=0}^{v}a_{rk}x^r y^k$$

设椭圆曲线E的基域为$GF(p)$，$G_l(x,y)$在基域$GF(p)$上与l次模多项式$\Phi_l(x,y)$具有相同的素因子分解形式，即在$G_l(x,y)$和$\Phi_l(x,y)$的素因子分解式中，两者所包含的素因子个数是相同的，经过适当地调整顺序后，对应因子的次数也相同。而$G_l(x,y)$有着

比 $\Phi_l(x,y)$小得多的系数，因此，在 SEA 算法的实现中，可以用 $G_l(x,y)$来替代 $\Phi_l(x,y)$。

对 $r=0,1,2,\cdots,l+1$，若令

$$s_r(\tau)=\sum_{k=0}^{v}a_{l+1-r,k}j(l\tau)^k \tag{4.9}$$

则由定理 4.2 可知，函数 $h(x)=\sum_{r=0}^{l+1}s_r(\tau)x^{l+1-r}$ 有根 $f\left(\tau+\frac{n}{l}\right)$ 和 $\frac{l^s}{f(l\tau)}$，式中，$n=0,1,\cdots,l-1$。

为了求解 $G_l(x,y)$中的系数 $a_{rk}(r=0,1,2,\cdots,l+1;k=0,2,\cdots,v)$，可将式(4.9)的两边分别展开为 τ 的 Fourier 展开式，然后比较等式两边同次项的系数即可得到 a_{rk}。为此，设 $q=e^{-2\pi i\tau}$，则 $f(\tau)^r$的 Fourier 展开式为

$$f(\tau)^r=q^{-vr}\left(\sum_{i=0}^{\infty}b_iq^i\right) \tag{4.10}$$

并记

$$\begin{cases}c_{r,1}(\tau)=\sum_{n=0}^{l-1}f\left(\tau+\frac{n}{l}\right)^r\\ c_r(\tau)=c_{r,1}(\tau)+\left(\frac{l^s}{f(l\tau)}\right)^r\end{cases} \tag{4.11}$$

则由式(4.10)，有

$$c_{r,1}(\tau)=q^{-vr}\left(\sum_{i=0}^{\infty}l_ib_iq^i\right) \tag{4.12}$$

式中

$$l_i=\begin{cases}l, \text{当 } i\equiv vr \bmod l\\ 0, \text{其他}\end{cases}$$

由式(4.11)和式(4.12)，可进一步展开 $c_r(\tau)$，进而由 Newton 公式可得到 $s_r(\tau)$的迭代递推公式为

$$\begin{cases} s_r(\tau) = -\sum_{i=0}^{r} c_{r-i}(\tau)s_i(\tau) \\ s_0(\tau) = 1 \end{cases} \tag{4.13}$$

类似地，利用模函数$j(\tau)$可将式(4.9)的右边展开为Fourier展开式。对确定的r，比较等式(4.9)两边Fourier展开式中同次项的系数即可得到$a_{l+1-r,k}$的值，进而可以求出模多项式$G_l(x,y)$的全部系数a_{rk}。具体算法如下。

算法4.3　模多项式算法

输入：素数l

输出：模多项式$G_l(x,y)$

1. 计算模函数$f(\tau)$和$j(l\tau)$；
2. $a_{l+1,0}\leftarrow 1, a_{l+1,k}\leftarrow 0, k=1,2,\cdots,v$；
3. $S_0(\tau)\leftarrow 1, C\leftarrow 1, D\leftarrow \sum_{i=0}^{\infty} l_i b_i q^i$；
4. For $r=1,2,\cdots,l+1$ do
   - 4.1　$C\leftarrow C\times\left(\frac{l^s}{f(l\tau)}\right)$；
   - 4.2　$D\leftarrow D\times q^{-v}$[由式(4.12)计算$c_{r-1}(\tau)$]；
   - 4.3　$c_r(\tau)\leftarrow D+C$[由式(4.11)计算$c_r(\tau)$]；
5. 由迭代式(4.13)计算$s_r(\tau)$；
6. For $k=v,v-1,\cdots,0$ do
   - 6.1　$a_{l+1-r,k}\leftarrow \hat{s}_r(\tau)$的$q^{-lk}$项次数；
   - 6.2　$s_r(\tau)\leftarrow s_r(\tau)-a_{l+1-r,k}j(l\tau)^k$；
7. 输出模多项式 $G_l(x,y)=\sum_{r=0}^{l+1}\sum_{k=0}^{v} a_{rk}x^r y^k$；

4.4 Elkies 算法及其实现

当$\gcd(x^p-x,\Phi_l(x,j(E)))\neq 1$时，$l$为Elkies 素数。Elkies 指出：若模多项式$\Phi_l(x,j(E))$在$GF(p)$上有零点g，则至少存在E的一个l阶子群C，其j不变量在$GF(p)$中使得C在Frobenius 映射φ下不动。也就是说，椭圆曲线E上存在一个定义于有限域$GF(p)$上的l次同种$I:E\to E/C$，使得I的核C是Frobenius 映射φ在$E[l]$上的一个一维特征子空间，故$\varphi(C)=C$。亦即存在$\lambda\in GF(l)$，$P\in E[l]$，使得$\varphi(P)=\lambda P$，$P\in C$。

现令

$$h_l(x)=\prod_{P\in C\setminus\{O\}}(x-x(P)) \tag{4.14}$$

式中，$x(P)$表示点P的x坐标。于是，$\deg(h_l(x))=\frac{l-1}{2}$。由$\varphi(C)=C$和$C\subset E[l]$可知，$h_l(x)$是可除多项式$f_l(x)$的一个因子。因$C$在Frobenius 映射$\varphi$下不动，故曲线$E$的$l$次可除多项式$f_l(x)$存在有理多项因子$h_l(x)\in F_p[x]$。因此，求解$h_l(x)$等价于对$E$的$l$次同种$I$的计算。为方便起见，也称$h_l(x)$为可除多项式。

利用$h_l(x)$，可采用与Schoof 算法中类似的穷举算法求解使同余方程

$$\varphi(P)\equiv[\lambda](P)\bmod h_l(x),P\in C \tag{4.15}$$

成立的λ，亦即Frobenius 映射φ的特征值。

最后，由$t\equiv\lambda+\frac{p}{\lambda}\bmod l$即可求出$t\bmod l$的值。

具体地说，Elkies 算法的工作过程如下。

① 计算同种映射的核，得到可除多项式的一个因子$h_l(x)$，且

$\deg h_l(x)=\dfrac{l-1}{2}$；

② 寻找合适的λ，使得方程(4.15)成立，该λ即为φ的特征值；

③ 由$t\equiv\lambda+\dfrac{p}{\lambda} \bmod l$即可求出$t \bmod l$。

本节将根据Elkies算法的需要，分别介绍求解同种曲线、可除多项式和特征值的具体实现方法。

1. 计算同种曲线参数，求解可除多项式

对椭圆曲线E上的定义于有限域$GF(p)$中的l次同种$I:E\to E/C$，有下面的定理。

定理4.3　设C是椭圆曲线E在Frobenius映射φ下不动的l阶子群，E/C为E的同种曲线，则同种映射$I:E\to E/C$具有如下形式。

$$P=(x,y)\to\left(\frac{k(x)}{h_l^2(x)},\frac{g(x,y)}{h_l^3(x)}\right) \tag{4.16}$$

现设素域上的椭圆曲线方程如式(2.14)，E/C的方程为$y^2=x^3+\tilde{a}x+\tilde{b}$，$h_l(x)=x^d+a_{d-1}x^{d-1}+\cdots+a_0$，$\deg h_l(x)=d=\dfrac{l-1}{2}$，则在已知椭圆曲线$E$的参数$(a,b)$以及模多项式$\Phi_l(x,j(E))$及其在$GF(p)$中的一个根$g$的条件下，可以利用算法4.4求出同种曲线的参数$(\tilde{a},\tilde{b})$以及$a_{d-1}$。

算法4.4　计算同种曲线的参数

输入：椭圆曲线E，模多项式$\Phi_l(x,j(E))$及s,g 输出：$(\tilde{a},\tilde{b})$和a_{d-1}
1. $E_4\leftarrow-\dfrac{a}{3},E_6\leftarrow-\dfrac{b}{2},\Delta\leftarrow\dfrac{E_4^3-E_6^2}{1728},\tilde{\Delta}\leftarrow\dfrac{\Delta g^{12/s}}{l^{12}},j\leftarrow\dfrac{E_4^3}{\Delta}$；

2. 计算 D_G, D_j

$$D_G=g\left(\frac{\partial}{\partial x}\Phi_l(x,y)\right)\Big|_{(g,j)}, D_J=j\left(\frac{\partial}{\partial y}\Phi_l(x,y)\right)\Big|_{(g,j)};$$

3. 若 $D_J=0$，则

3.1　$\tilde{E}_4=\frac{E_4}{l^2}, \hat{a}=-3l^4\tilde{E}_4, \tilde{j}=\frac{\tilde{E}_4^3}{\tilde{\Delta}}$；

3.2　输出 $(\tilde{a},\tilde{b})=\hat{a}\pm 2l^6\sqrt{\tilde{\Delta}(\tilde{j}-1728)}, a_{d-1}=0$，并退出；

4. $E_2\leftarrow -\frac{12E_6D_J}{sE_4D_G}, E_0\leftarrow\frac{E_6}{E_4E_2}$；

5. $g'\leftarrow -\frac{sE_2g}{12}, j'\leftarrow -\frac{E_4^2E_6}{\Delta}$；

6. 计算 D'_G, D'_J：

$$D'_G=g'\left(\frac{\partial}{\partial x}\Phi_l(x,y)\right)\Big|_{(g,j)}+gg'\left(\frac{\partial^2}{\partial x^2}\Phi_l(x,y)\right)\Big|_{(g,j)}$$
$$+gj'\left(\frac{\partial^2}{\partial x\partial y}\Phi_l(x,y)\right)\Big|_{(g,j)};$$

$$D'_J=j'\left(\frac{\partial}{\partial y}\Phi_l(x,y)\right)\Big|_{(g,j)}+jj'\left(\frac{\partial^2}{\partial y^2}\Phi_l(x,y)\right)\Big|_{(g,j)}$$
$$+jg'\left(\frac{\partial^2}{\partial y\partial x}\Phi_l(x,y)\right)\Big|_{(g,j)};$$

7. $E'_0\leftarrow\left(-\frac{s}{12}D'_G-E_0D'_J\right)D_J^{-1}$；

8. $\tilde{E}_4\leftarrow[E_4-E_2(12E'_0E_0^{-1}+6E_4^2E_6^{-1}-4E_6E_4^{-1})+E_2^2]l^{-2}$；

9. $\tilde{j}\leftarrow\frac{\tilde{E}_4^3}{\tilde{\Delta}}, f\leftarrow\frac{l^s}{g}, f'\leftarrow\left(\frac{s}{12}\right)E_2f$；

10. $\tilde{E}_6 \leftarrow \dfrac{\tilde{E}_4 f'}{l \times \tilde{j}} \left(\dfrac{\partial}{\partial x} \Phi(x,y) \right) \Big|_{(f,\tilde{j})} \left(\dfrac{\partial}{\partial y} \Phi(x,y) \right) \Big|_{(f,\tilde{j})}^{-1}$；

11. 输出$(\tilde{a},\tilde{b}) = (-3l^4 \tilde{E}_4, -2l^6 \tilde{E}_6)$，$a_{d-1} = -\dfrac{l}{2} E_2$。

在算法4.4中，由于已知$j(E)$，所以模多项式$\Phi_l(x, j(E))$在$GF(p)$中的有理根g可以通过求解代数方程的方法得到。

接着，利用算法4.5求解可除多项式$h_l(x)$。

算法4.5　可除多项式求解算法

输入：椭圆曲线E，同种曲线参数$(\tilde{a},\tilde{b})$以及a_{d-1}

输出：$h_l(x)$

1. $d \leftarrow \dfrac{l-1}{2}$，$a_d \leftarrow 1$；

2. For $k=1,2,\cdots,d$ do

   2.1　利用迭代式(4.17)和参数(a,b)计算c_k；

   2.2　利用迭代式(4.17)和参数$(\tilde{a},\tilde{b})$计算$\hat{c}_k$；

3. $wp_1(z) \leftarrow \dfrac{1}{z^2} + \sum\limits_{k=1}^{d} c_k z^{2k}$；

4. For $j=2,3,\cdots,d$ do

   4.1　$wp_j(z) \leftarrow wp_{j-1}(z) \times wp_1(z)$；

5. $g(z) \leftarrow z^{l-1} \sum\limits_{r=0}^{d} \dfrac{1}{r!} \left[\sum\limits_{k=1}^{d} \dfrac{l c_k - \tilde{c}_k}{(2k+1)(2k+2)} z^{2k+2} - a_{d-1} z^2 \right]^r$；

6. For $j=d-1,\cdots,0$ do

   6.1　$g(z) \leftarrow g(z) - a_{j+1} wp_{j+1}(z)$；

   6.2　$a_j \leftarrow g(z)$的最低次项的系数；

7. 输出$h_l(x) = x^d + a_{d-1}x^{d-1} + \cdots + a_0$。

上述算法中的系数 $c_k, \hat{c}_k$ 由下面的迭代公式决定，其中 $k>2$。

$$\begin{cases} c_1 = -\dfrac{a}{5} \\ c_2 = -\dfrac{b}{7} \\ \quad \vdots \\ c_k = -\dfrac{3}{(k-2)(2k+3)} \sum\limits_{j=1} c_j c_{k-1-j} \end{cases} \tag{4.17}$$

由前文可知，用于定义椭圆曲线有限群的有限域基域可简单地划分为两大类：一类是素域 $GF(p)$；另一类是 Galois 复合域 $GF(p^n)$。这两种基域在应用 SEA 算法计算椭圆曲线有限群的阶时，除了计算 l 次同种外，没有什么大的区别。前面已经讨论了关于素域上的 l 次同种和特征多项式的计算。而关于复合域 $GF(p^n)$ 上的 l 次同种的计算，Couveignes、Lercier 等人于1994—1996 年间提出了一系列的理论和算法，较成功地解决了这一问题。限于篇幅和本书研究的需要，这里不再讨论。

2. 确定Frobenius 映射 φ 的特征值 λ，完成Elkies 算法

当 l 为 Elkies 素数时，为求解使得方程式(4.15)成立的 Frobenius 映射 φ 在 $GF(l)$ 中的一个特征值 λ，因为

$$\varphi(P) \equiv [\lambda](P) \bmod h_l(x), P \in C$$

所以有

$$(x^p, y^p) \equiv [\lambda](x, y) \bmod h_l(x) \tag{4.18}$$

以及以下定理。

定理 4.4 设素域上的椭圆曲线 E 的方程如式(2.14)，(x, y) 是曲线 E 上的任一点，m 是整数，则有

$$[m](x, y) = \left(x - \frac{\Psi_{m-1}\Psi_{m+1}}{\Psi_m^2}, \frac{\Psi_{m+2}\Psi_{m-1}^2 - \Psi_{m-2}\Psi_{m+1}^2}{4y\Psi_m^2} \right) \tag{4.19}$$

式中，Ψ_i 可由下面的迭代式得到。

$$\begin{cases}\Psi_{2m+1}=\Psi_{m+2}\Psi_m^3-\Psi_{m+1}^3\Psi_{m-1}, & (m>1)\\ \Psi_{2m}=\dfrac{\Psi_m(\Psi_{m+2}\Psi_{m-1}^2-\Psi_{m+1}^2\Psi_{m-2})}{2y}, & (m>2)\end{cases} \tag{4.20}$$

迭代初始值如下：

$$\begin{cases}\Psi_{-1}=-1\\ \Psi_0=1\\ \Psi_1=2\\ \Psi_2=2y\\ \Psi_3=3x^4-6ax^2+12bx-a^2\\ \Psi_4=2y(2x^6+10ax^4+40bx^3-10a^2x^2-8abx-16b^2-2a^3)\end{cases}$$

这样，对所有的 $\lambda\in GF(l)$，通过利用定理 4.4 来验证方程式(4.18)是否成立，可确定Frobenius 映射 φ 的特征值 λ，接着计算 $t\equiv\lambda+\frac{p}{\lambda} \bmod l$，可求出 $t \bmod l$。这就是Elkies 方法，具体算法如下。

算法 4.6　Elkies 算法的第二阶段

输入：椭圆曲线 E，可除多项式 $h_l(x)$
输出：$t \bmod l$

1. 计算 $\Psi_{-1}(x,y)\sim\Psi_4(x,y)$；
2. For $m=5$ to $\dfrac{l+3}{2}$
   2.1　利用迭代式(4.20)，计算 $\Psi_m(x,y)$；
3. $l_x\leftarrow x^p \bmod h_l(x)$；
   $l_y\leftarrow y(x^3+ax+b)^{\frac{p-1}{2}} \bmod h_l(x)$；
4. For $\lambda=1$ to $\dfrac{l-1}{2}$

4.1 $r_\lambda \leftarrow (x\Psi_\lambda^2 - \Psi_{\lambda-1}\Psi_{\lambda+1}) \bmod h_l(x)$;

4.2 若 $l_x\Psi_\lambda^2 \equiv r_x \bmod h_l(x)$，则

(a) $r_y \leftarrow \Psi_{\lambda+2}\Psi_{\lambda-1}^2 - \Psi_{\lambda-2}\Psi_{\lambda+1}^2$;

(b) 若 $r_y \equiv 4yl_y\Psi_\lambda^3 \bmod h_l(x)$，则跳出循环;

(c) 若 $-r_y \equiv 4yl_y\Psi_\lambda^3 \bmod h_l(x)$，则 $\lambda \leftarrow -\lambda$，跳出循环;

5. 返回 $t \equiv \lambda + \frac{p}{\lambda} \bmod l$。

关于Elkies 算法，Morain 等人做了进一步的改进。他们指出：对于给定的Elkies 素数 l，$t^2-4p \bmod l$ 在有限域 $GF(l)$ 中是一个平方剩余，l 次模多项式 $\Phi_l(x, j(E))$ 在有限域 $GF(q)$ 上有两个不同的根。相应地，Frobenius 映射 φ 也有两个不同的特征值 τ_1, τ_2。与之对应，E 有两个 l 次同种映射 I_1, I_2 和相应的两条同种曲线 E_1，E_2，记为：$E \xrightarrow{I_1} E_1$，$E \xrightarrow{I_2} E_1$。

相应地，E_1 也有两个 l 次同种映射。可以证明，其中一个必定是 I_1 的对偶映射 I_1^*，另一个一般记作 I_{11}。由此，可以得到一条由特征值 τ_1 所决定的同种曲线链：$E \xrightarrow{I_1} E_1 \xrightarrow{I_{11}} E_{11}$。

继续这一过程，可以得到分别由特征值 τ_1, τ_2 所确定的两条同种曲线链，如下所示。

$$E \xrightarrow{I_1} E_1 \xrightarrow{I_{11}} E_{11} \xrightarrow{I_{111}} E_{111} \rightarrow \cdots$$

$$E \xrightarrow{I_2} E_2 \xrightarrow{I_{22}} E_{22} \xrightarrow{I_{222}} E_{222} \rightarrow \cdots$$

因 $E, E_1, E_{11}, \cdots$ 都是定义在 $GF(p)$ 上的曲线，由 $GF(p)$ 的有限性可知，在同构意义下，$E, E_1, E_{11}, \cdots$ 必定是一个周期序列。所以，上面所得到的两条同种曲线链实际上是两个定义在有限域 $GF(p)$ 上的 l 次同种圈。

这样，对某个不太大的确定的整数k，可以通过计算$t \bmod l^k$的可除多项式$f_{l^k}(x)$的一个因子$h_{l^k}(x)$，进而求出$t \bmod l^k$。这里有

$$\deg(h_{l^k}(x)) = \frac{(l-1)l^{(k-1)}}{2}$$

也就是说，利用上述的方法，可以将可除多项式$f_{l^k}(x)$转换为一个次数不超过$\frac{(l-1)l^{(k-1)}}{2}$的多项式，这就是Morain同种圈算法。虽然它并不能降低Elkies算法的时间复杂度，但能够在一定的程度上加快运算速度，减少运行时间，因此成为SEA算法的重要改进之一。

只有当$t^2-4p \bmod l$在有限域$GF(l)$中是一个平方剩余时，Elkies算法及其改进的Morain同种圈算法才是有效的。在全部素数中，大约有一半的素数是Elkies素数，当l不是Elkies素数时，则需要使用在下一节讨论的Atkin算法。

4.5　Atkin算法及其实现

Atkin算法的基本思想最初于1988—1991年在Internet的电子邮件讨论组上发表，目前已无法找到。但由于Atkin本人从未正式公开发表过他的Atkin改进算法，所以本节的内容是在整理散见于各文献中关于Atkin改进算法的资料后综合而成的。

当$\gcd(x^{p^i}-x, \Phi_l(x, j(E)))=1$，$l$为Atkin素数时，特征方程$f_l(x)=0$无有理根，$t^2-4p$在有限域$GF(l)$是一个非二次剩余，这时，Atkin算法过程如下。

首先对$i=1,2,3\cdots$逐一验证

$$\gcd(x^{p^i}-x, \Phi_l(x, j(E))) = \Phi_l(x, j(E)) \qquad (4.21)$$

直至方程式(4.21)成立为止。这一步等价于寻找一个使$\Phi_l(x,j(E))$能够在$GF(p^i)$中完全分裂的最小的i,亦即定理4.1中的r。

设t是Frobenius映射的迹,则t满足

$$t^2 \equiv p\left(\xi + \frac{1}{\xi} + 2\right) \bmod l \tag{4.22}$$

式中,$\xi \in \overline{F}_l$是某个r次本原单位根。

因此,只需要求出$GF(l)$中全部r次本原单位根ξ,将它们带入式(4.22),即可求出$t \bmod l$的一个可能值的集合T_l。

具体算法如下。

算法4.7　Atkin算法

输入:$l,\Phi_l(x,j(E))$
输出:(l,T_l)

1. $T_l \leftarrow \{\ \}$,$drest \leftarrow l+1$;
2. 若$p=l$,则对所有的d,
   若$d \mid drest$,且$\frac{(d-1)drest}{2}$是偶数,则转至第4步;
3. 若$p=-l$,则对所有的d,
   若$d \mid drest$,且$\frac{(d-1)drest}{2}$是奇数,则转至第4步;
4. 若$d_2=l+1$,则$d_2 \leftarrow \deg(\gcd(x^{p^d}-x,\Phi_l(x,j(E))))$;
5. 计算$F_l^* = F_l[\sqrt{d}]^*$的一个生成元g;
6. $S \leftarrow \{g^{i(l^2-1)/r} : (i,r)=1\}$;
7. 对每一个$\xi \in S$:
   7.1　$t \leftarrow p\left(\xi+\frac{1}{\xi}+2\right) \bmod l$;

7.2 $t \leftarrow \sqrt{t} \bmod l$;

7.3 $T_l \leftarrow T_l \cup \{t, -t\}$;

8. 返回(l, T_1)。

从理论上讲,Atkin 算法不是一个很好的算法,但由于它能弥补 Elkies 算法的不足,以及和 Elkies 算法结合起来投入实际应用时,能收到非常好的效果,因此,Atkin 改进也是非常有意义的。

4.6 SEA 算法的最后步骤

到此为止,已经求出了所有的 $t \bmod l$ 的集合,在本节中将讨论如何利用这些信息确定椭圆曲线有限群的阶 $\#E(GF(p))$。

现在假定已经获得了足够多的素数 l(即满足 $\prod l > 4\sqrt{q}$ 时),它们构成集合 L。当 l 是Elkies 素数时,$t \bmod l$ 的精确值已知;而当 l 是 Atkin 素数时,有一个 $t \bmod l$ 的可能值的集合。记

$$\begin{cases} L_E = \{l \in L \mid l \text{ 为 Elkies 素数}\} \\ L_A = \{l \in L \mid l \text{ 为 Atkin 素数}\} \\ T_l = \{t \bmod l, l \in L_A\} \end{cases}$$

以及

$$m_3 = \prod_{l \in L_E} l$$

对所有的Elkies 素数 $l \in L_E$,由中国古代剩余定理可得一满足 $t \equiv t_3 \bmod m_3$ 的最小正整数 t_3,它们构成集合 S_3。

对于Atkin 素数集合 L_A,将其分成两个大致相等的集合 L_{A1} 和 L_{A2},使得整数 $\prod_{l \in L_{A1}} |T_l| \approx \prod_{l \in L_{A2}} |T_l|$,并记 $m_1 = \prod_{l \in L_{A1}} l, m_2 = \prod_{l \in L_{A2}} l$。

对每一个$l \in L_{A1}$，取定T_l中的一个元素t_l，对L_{A1}中的其他素数及相应的t'_l，应用中国古代剩余定理后可得一满足$t \equiv t_1 \bmod m_1$的最小正整数t_1。现使t_l遍历T_l，可得由t_1的所有不同取值构成的集合S_1。对于集合L_{A2}，类似地，可得由满足$t \equiv t_2 \bmod m_2$的所有t_2构成的集合S_2。

显然，m_1，m_2和m_3两两互质，且$m_1 m_2 m_3 = \prod_{l \in L} l$。

任取$t_1 \in S_1$，$t_2 \in S_2$，对由(t_1, m_1)，(t_2, m_2)和(t_3, m_3)构成的同余方程组

$$\begin{cases} t \equiv t_1 \bmod m_1 \\ t \equiv t_2 \bmod m_2 \\ t \equiv t_3 \bmod m_3 \end{cases} \tag{4.23}$$

由中国古代剩余定理及Hasse定理可得

$$t = t_3 + m_3(r_1 m_2 + r_2 m_1) \tag{4.24}$$

式中，

$$\begin{cases} r_1 \equiv \dfrac{t_1 - t_3}{m_2 m_3} \bmod m_1 \\ r_2 \equiv \dfrac{t_2 - t_3}{m_1 m_3} \bmod m_2 \end{cases} \tag{4.25}$$

对所有的$t_1 \in S_1$，$t_2 \in S_3$和$t_3 \in S_3$，式(4.25)将给出不同的r_1，r_2数对。由式(4.24)可得到所有可能的t值，而其中仅有唯一的一个是正确的t值。为了迅速地确定正确的t值，Atkin于1988年提出了利用大步小步算法来实现加速搜索。这是目前SEA算法在最后阶段的标准算法。

由于椭圆曲线有限群的阶满足$\#E(GF(p)) = q + 1 - t$，所以，对于椭圆曲线有限群中的任意一点$P \in E(GF(p))$，若t是正确的，则应有

$$(q + 1 - t)P = O \tag{4.26}$$

将式(4.24)代入式(4.26)可得

$$(q+1-t_3)P-r_1m_2m_3P=r_2m_1m_3P \tag{4.27}$$

在式(4.25)中，若取$t_3\in[0,m_3)$，且$|r_1|\leqslant\frac{m_1}{2}$，容易证明$|r_2|\leqslant m_2$。

这样，可以按如下方式应用大步小步算法。

第一步(小步)：首先对于每一个$t_1\in S_1$，由式(4.25)求出满足$|r_1|\leqslant\frac{m_1}{2}$条件的$r_1$，然后计算$(q+1-t_3)P-r_1m_2m_3P$，并将其和对应的$r_1$存储在表$H$中。

第二步(大步)：依次对所有的$t_2\in S_2$，通过式(4.25)计算r_2以及$r_2m_1m_3P$，并在表H中查找$r_2m_1m_3P$。若$r_2m_1m_3P\in H$，则得到了一组满足方程式(4.27)的(r_1,r_2)，从而由式(4.24)可求得t值，进而可求出$h=q+1-t$。

但是，对点$P\in E(GF(p))$，这里所求出的h实际上只是点P的阶的某一倍数。为了验证h是椭圆曲线有限群E的阶，还需另取其他若干$P'\in E(GF(p))$，验证$hP'=O$是否成立。

具体算法如下。

算法4.8　SEA算法的最后步骤

输入：$t \bmod l$的集合S_1，S_2和S_3

输出：$\#E(GF(p))$

1. 任取$P\in E(GF(p))$，$H\leftarrow\{\ \}$；
2. $Q_0\leftarrow(p+1-t_3)P$，$Q_1\leftarrow m_2m_3P$；
3. $Q_3\leftarrow m_1m_2m_3P$，$Q_2\leftarrow m_1m_2P$；
4. 对所有的$t_1\in S_1$：

4.1 $r_1 \leftarrow \frac{t_1 - t_3}{m_2 m_3} \bmod m_1$，且使 $|r_1| \leqslant \frac{m_1}{2}$；

4.2 $Q \leftarrow Q_0 - r_1 Q_1$；

4.3 $H \leftarrow H \cup \{Q, r_1\}$；

5. 对所有的 $t_2 \in S_2$ 的一个生成元 g：

5.1 $r_2 \leftarrow \frac{t_2 - t_3}{m_1 m_3} \bmod m_2$，且 $-m_2 < r_2 < 0$；

5.2 $H_1 \leftarrow r_2 Q_2$，$H_2 \leftarrow H_1 + Q_3$；

5.3 若 $(H_1, r_1) \in H$，则

(a) $h \leftarrow p + 1 - t_3 - (r_1 m_2 m_3 + m_1 r_2 m_3)$；

(b) 转至第5.6步；

5.4 若 $(H_2, r_1) \in H$，则

(a) $h \leftarrow p + 1 - t_3 - (r_1 m_2 m_3 + m_1 r_2 m_3 + m_1 m_2 m_3)$；

(b) 转至第5.6步；

5.5 继续下一轮循环；

5.6 另取 n 个 $P \in E(GF(p))$；

5.7 若对所有的 P，$hP = O$ 均成立，则返回 $\#E(GF(p)) = h$，并退出；否则，继续下一轮循环。

4.7 SEA 算法的实现

由SEA 算法的框架可知，SEA 算法的工作过程有如下三个阶段。

① 在基域 $GF(p)$ 上构造模多项式的集合 $\{G_l(x, y)\}$，其中 $G_l(x, y)$ 在 $GF(p)$ 上与 l 次模多项式 $\Phi_l(x, y)$ 具有相同的素因子分

解形式。

② 根据模多项式的分裂形式等条件，利用Elkies方法、Atkin方法和同种圈方法等，收集所有的Frobenius映射迹t模一系列小素数的信息。

③ 综合所收集到的信息，利用中国古代剩余定理，获取t的候选值集合T，再利用大步小步算法，从T中确定Frobenius映射迹t的确切值，进而计算椭圆曲线有限群E的阶$\#E(GF(p))$。

第①阶段为预处理阶段。可以证明，对于基域$GF(p)$，SEA算法中所用到的最大素数不超过$\lg p$。故在这一阶段，需要准备一个不超过$\lg p$的模多项式列表，供后继算法使用，其时间复杂度与后继阶段无关。

由于第①阶段的时间消耗与第②、③阶段相对独立，而第③阶段的时间消耗与候选值集合T中元素的个数$\#T$成正比，所以，为了提高SEA算法的运行效率，必须在第②阶段适当地控制$\#T$的大小。

在第②阶段，有Elkies方法、Atkin方法和同种圈方法可供选择，三种方法各有其优、缺点及应用条件，关键的计算是根据模多项式的分裂形式判断小素数l的类型。

对于Elkies素数，可以选用Elkies方法或同种圈方法。虽然同种圈算法优于Elkies方法，但由于同种圈方法只适用于较小的素数，所以，为了提高算法的运行速度，可设定一个阈值I_m，只对小于I_m的Elkies素数，选用4.4节所讨论的改进同种圈方法。

对于Atkin素数，Atkin方法可以较快地找到$t \bmod l$的一个可能值的集合T_l，但当集合T_l中的元素个数$\#T_l$较大时，会导致$\#T$的迅速增加，从而增加了第③阶段的时间花费。为了有效地控制$\#T$的大小，可以设定一个有关$\#T_l$的阈值A_m，当$\#T_l$超过A_m时则抛弃该Atkin素数，选择下一个新素数，继续尝试。

由于第①阶段为预处理阶段，需要预先计算出一批模多项式供后继阶段使用，那么需要预先计算多少才合适呢？

1852 年，俄罗斯数学家切比雪夫证明了如下的素数分布不等式。

定理 4.5 设$\pi(x)$表示不大于x的素数个数，存在两个正常数c_1、c_2，使得下列切比雪夫素数分布不等式成立。

$$c_1 \frac{x}{\ln x} \leqslant \pi(x) \leqslant c_2 \frac{x}{\ln x} \tag{4.28}$$

之后，他又进一步指出，切比雪夫不等式中的$c_1=1$，$c_2=1.2$。

考虑区间$[e^k, e^{k+1}]$中的素数个数C_k，由素数分布不等式(4.28)可知

$$\begin{aligned} C_k &= \pi(e^{k+1}) - \pi(e^k) \\ &\leqslant c_2 \frac{e^{k+1}}{(k+1)\ln e} - c_1 \frac{e^k}{k \ln e} \\ &= \frac{c_2 e^{k+1}}{k+1} - \frac{c_1 e^k}{k} \\ &< \frac{c_2 e^{k+1}}{k+1} \end{aligned}$$

所以，区间$[e^k, e^{k+1}]$中所有素数的乘积$M_k = \prod l < (e^{k+1})^{C_k} = e^{(k+1)C_k}$，故有$\ln M_k < C_k(k+1) < c_2 e^{k+1}$。这样，对所有不大于$L$的素数，其乘积$M$满足

$$\ln M < \sum_{k=2}^{[\ln L]+1} c_2 e^k < c_2 \left(\frac{eL-1}{e-1} \right) \tag{4.29}$$

另一方面，由SEA 算法的要求，素数乘积M还应满足

$$M > \prod l > 4\sqrt{p} \tag{4.30}$$

联合式(4.29)和式(4.30)，并代入$c_2=1.2$，可知$L \geqslant \frac{1}{2}\ln p$。

考虑到有部分 Atkin 素数可能被抛弃，而所有的素数中，

Elkies 素数出现的概率为1/2，所以，取$L=\ln p$作为在预处理阶段所需要处理的模多项式次数的上限。例如，对素域$GF(p)=GF(2^{160}-47)$，只需对不超过$L=\ln(2^{160}-47)=111$的素数进行求解模多项式的操作即可。

而有关本决策方案的两个阈值I_m和A_m的选择，则需要根据基域$GF(p)$的情况和实践经验来决定。这里，选定$I_m=5$，$A_m=18$，并用Borland 公司的Delphi 6.0加嵌入式汇编来实现。

对于素数$p=2^{160}-47$，系统运行的结果如下。

① 有限域的阶

$p=2^{160}-47$

$=1461501637330902918203684832716283019655932542929$

② 随机选取的曲线参数

$a=150109981375978573244826447265299164247145 85$

$b=213910743184753764084596661683117507834307 82$

③ 曲线的阶

$r=1461501637330902918203684313125075552876176419731$

④ 分解结果

$r=11\times 8934988813931920809 1$

$\times 1487005613311253628184160731$

第5章　椭圆曲线密码体系

椭圆曲线密码体系(Elliptic Curve Cryptosystem,ECC)是基于椭圆曲线离散对数问题的一类新型公钥密码体系,基本上可以看成是对有限域乘法群上离散对数问题的各种密码体系的重新实现。但由于椭圆曲线有限群的自身有一些特殊结构,故椭圆曲线密码体系与其他基于有限域乘法群上的密码体系又有许多细微的不同。

在前面的几章中,从数学理论上先后讨论了椭圆曲线公钥密码学的数学基础、离散对数问题和数点问题。本章在前面几章的基础上,首先讨论有关通信协议的安全性问题,然后根据安全通信的需要,给出作者所设计的全新的、安全高效的密钥管理、信息加密、数字签名等多种椭圆曲线密码体系方案,及对这些方案的分析。

5.1　密码协议及其安全性

协议(Protocol)是日常生活中经常用到的一个概念。所谓协议,就是指两个或者两个以上的参与者为了完成某一特定任务而采取的一系列的步骤。在这个定义中,协议包含了以下三层含义。

① 协议自始至终是一个有序的过程,每一步骤必须按照约定依次执行。在上一步还没有执行完毕之前,后面的步骤是不可能执行的。

② 对一个协议而言，它至少需要两个参与者，否则就不构成协议。

③ 协议具有明确的目的，一个协议的执行是为了完成某项确定的任务，没有目的的任务是构不成协议的。

类似地，称具有通信功能的协议为通信协议。也就是说，通信协议是指为了完成某种通信任务和数据交换的协议。

一般来说，通信协议应该具备以下一些特点。

① 协议的每一步必须被确切地定义；

② 通信协议中的每一步要么是由通信的一方或多方进行某种运算，要么是在通信的各参与方之间传输消息，二者必居其一；

③ 必须能对每一种可能发生的情况作出反应；

④ 通信的各参与方必须知道协议的每一个步骤；

⑤ 通信的各参与方都必须同意完全遵守该协议。

简单地说，通信协议必须具备有效性、完整性和公平性。

在本书的研究中，主要关注的是通信过程的安全问题，研究的是基于密码技术的通信安全解决方案。所以，在这里借鉴了王育民教授的观点，将通信协议分成普通协议和密码协议两种类型，并将上述的这类信息安全解决方案统称为密码协议。具体地说，密码协议是指：建立在密码体系的基础之上，运行在计算机通信网络和分布式系统中，使用密码技术来保证通信过程中的信息安全，完成密钥管理、身份认证、数据保密等任务的通信协议。

从上面的定义可以看出，密码协议包括两个基本要素。

① 使用了密码技术；

② 能够对信息提供安全防护。

在这两个要素中，使用密码技术是相对容易的，但使用了密码技术未必就能实现预定的安全目标。因此，通常所说的密码协议都是指那些使用了密码技术并试图保护在通信过程中信息安全的协

议。

判断一个密码协议设计得是否严谨、能否达到其预定的安全目标，就涉及密码协议的安全性问题。对于一个密码协议，如果它能够使任何非法的攻击者都不可能从协议中获得比协议自身所体现的信息更多的有效信息，则称该协议是安全的。这样，一个安全的密码协议意味着能够实现预定的安全目标。而分析、研究一个密码协议使其能够实现预定的安全目标的过程则称为对该密码协议的安全性分析。

至今为止，虽然还没有一种可靠的方法可以证明某一密码协议是真正安全的，但经过人们的不懈努力，还是有一些方法可以用于发现一个待分析的密码协议中的安全漏洞，从而证明该密码协议是不安全的。

本节将综合讨论在密码协议的安全性分析和论证所取得的成果，为后面的相关协议的设计和分析做准备。

5.1.1 密码协议分析的前提

为了方便后继的讨论，这里先对密码协议分析的前提做一些约定。

1. 完善加密前提

密码协议的安全性分析关注的是密码协议本身，而不考虑所使用的密码技术本身以及管理方面的安全性问题。具体地说，这些前提包括以下几方面。

① 密码协议所采用的密码技术是完美的，不考虑密码技术本身被攻破的情况。

② 解密密钥(私有密钥或单钥)只为合法的拥有者所拥有，不

考虑密钥被盗的情况。

③ 只有解密密钥才能解密所收到的加密数据，不考虑密钥雷同问题。

④ 无加密项冲突，即若有$\{m_1\}_{K_1}=\{m_2\}_{K_2}$，则必有$m_1=m_2$以及$k_1=k_2$成立。

2. 协议的参与者

根据其目的不同，参与协议运行的主体可以分为诚实的合法用户和攻击者。其中约定下列前提。

① 诚实的合法用户将会严格地依照协议的规定参与协议的运行，不考虑错误操作的情况。

② 攻击者可能是系统的合法用户，甚至可能就是参与通信的某一方，拥有自己的密钥。

③ 攻击者不会按照协议的规定参与协议的运行，否则就是诚实的合法用户了。

④ 攻击者企图获取他所不该知道的秘密或不该得到的利益，或者假冒其他诚实的合法用户。

3. 攻击者的知识与能力

在密码协议的安全性分析中，一般依据如图1-7所描述的由Dolev和Yao所提出的网络安全模型。同时假定网络通信信道为攻击者所控制，而且攻击者具有足够的知识和能力。其具体如下。

① 熟悉现代密码学，熟悉密码算法的工作原理以及加密、解密等运算和操作；

② 知道网络通信的参与各方的情况、主体名及其公钥，并拥有自己的加密密钥(公钥或单钥)和解密密钥(私钥或单钥)；

③ 具有无限存储能力，每当窃听到或收到一条信息或消息，

即可被存储下来，以备以后使用，同时还增加了攻击者的知识；

④ 可窃听及中途拦截网络通信中传递的任何信息；

⑤ 可解密所有用攻击者自己的加密密钥所加密的消息；

⑥ 可以破坏、篡改网络信道中所传递的信息；

⑦ 可以向网络信道中插入自己创建的信息；

⑧ 即使不知道所截获的信息中的加密部分的内容，攻击者也可以重放他所看到的任何信息，并能够修改其中的未加密部分；

⑨ 可以随意运用他所知道的任何知识，并拥有足够强大的资源。

图5-1 所示为一个简单的三方密码协议的运行框架。显然，攻击者完全控制了密码协议的三个参与方之间的信息交换，较好地反映了上面所假设的攻击者所具备的强大能力。

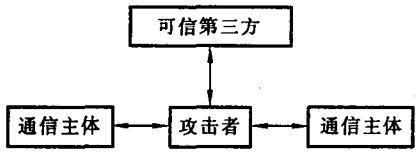

图 5-1　三方密码协议分析框架

5.1.2　密码协议分析的方法

正如前文所提到的，证明一个密码协议的不安全性要比证明其安全性容易得多。目前，虽然尚未出现一种能可靠地证明某一密码协议是真正安全的密码协议分析方法，但人们还是提出了许多可以用来发现协议中安全漏洞的方法。一般来说，按照密码协议安全性分析中是否采用了形式化分析工具，可以将密码协议分析方法分成形式分析方法和非形式分析方法两大类。

非形式分析方法又称为攻击检验方法，它利用各种已知的攻

击技术对所要分析的密码协议进行攻击，根据攻击的结果来检验所研究的密码协议是否安全。但实际上，肯定存在着许多目前未知的攻击方法，这使得此分析方法只能分析协议是否存在目前已知的安全缺陷，而不可能对其进行全面客观的分析，容易导致一个不安全的协议被误认为是安全的协议。另一方面，虽然这类分析方法不能证明协议是安全的，但由于它非常简单实用，能够较快地发现协议的缺陷，因此还是得到了较广泛的应用，一般用于对协议进行初步分析。

形式化分析方法希望通过将密码协议形式化，然后通过人工或者计算机辅助手段对所得到的密码协议的形式化表达进行分析，判断该密码协议是否可靠。与非形式分析方法相比，它不仅能够发现现有的已知的攻击方法对所研究的协议带来的威胁，而且还能够发现密码协议中的一些细微漏洞，找到对密码协议的新的攻击方法。

对于形式化分析方法，可以根据其设计思想，分成逻辑化方法、模型检测方法、定理证明方法和通用形式方法四种类型。

1. 逻辑化方法

逻辑化方法使用基于信任关系的逻辑系统，通过建立密码协议簇的需求关系的逻辑形式化模型对密码协议进行分析，是至今为止最为有效的，使用最为广泛的密码协议安全性分析方法。其基本思想体现在以下方面。

① 针对密码协议的背景给出若干基本的逻辑公理和相关规则；

② 针对密码协议的背景给出若干基本的分析前提，称为初始假设；

③ 将密码协议的操作步骤形式化为一系列的逻辑公式，称为

协议理想化；

④ 将密码协议所要实现的目标形式化为目标逻辑表达式；

⑤ 根据初始假设和逻辑公理，使用相关的逻辑推导规则进行推理。

如果能够从初始假设推理出目标逻辑表达式，则认为该密码协议是安全的；否则，认为协议是不安全的，存在着安全漏洞。因此，逻辑化方法又被称为认证逻辑。

最早的BAN逻辑化方法是由美国DEC公司的研究人员Michael Burrows、Martin Abadi和Roger Needham于1989年提出。BAN逻辑成功地对Needham-Schroeder认证协议、Kerberos认证协议等著名协议进行了分析，开辟了密码协议形式化分析的新方向，促成了其他更多的密码协议形式化分析方法的诞生，成为“密码协议形式化分析学之父”。

除了BAN逻辑以外，目前常用的逻辑化方法还有Kailar逻辑、GNY逻辑、AT逻辑、vO逻辑、SvO逻辑、WK逻辑、MB逻辑和AUTOLOG逻辑等不下十几种逻辑化分析方法。由于这些逻辑化方法都是基于BAN逻辑的工作模型，通过对BAN逻辑进行改进而得到的，所以，统称为BAN类逻辑。

2. 模型检测方法

模型检测方法是一种基于模型检测技术的形式化密码协议分析方法。它首先构造一个密码协议运行时的有限状态系统模型，然后利用有限状态系统的自动化检测分析工具来分析密码协议的安全性。由于模型检测方法对密码协议的自动验证和工程化设计具有较大的指导意义，因而得到了学术界越来越多的关注。

目前，模型检测方法已经被证明是一个非常有用的检测密码协议安全漏洞的工具，成功地发现了许多以前未被发现的新的攻

击。但模型检测方法仍然存在着许多问题，其中最大的问题在于如何将密码协议的运行状态构造成一个合适的有限状态系统模型，而又没有增加或减少待分析密码协议的安全性。

3. 通用形式方法

通用形式方法采用最弱前置谓词和Petri网等一些通用的形式化方法来说明和分析密码协议的安全性。

(1) 最弱前置谓词WP法

最弱前置谓词WP是证明程序正确性的有力工具。Yasinsac和William提出将密码协议看成一个计算机程序，利用最弱前置谓词WP对该程序进行分析，以求能够分析和检验该密码协议的正确性。然而，密码协议的正确性与安全性并不等价，也就是说，证明了一个密码协议的正确性并不等于其安全性得到了证明。因此，最弱前置谓词WP法实际上并不能检测出密码协议所存在的安全缺陷。

(2) Petri网法

Petri网法将Petri网作为认证协议的表示工具，通过对Petri网进行分析来证明密码协议的安全性。

虽然上述这些通用形式方法已经被应用于密码协议的分析与检测之中，但这类方法没有很好的理论研究基础，只能就事论事，不能说明某一密码协议是否真正安全。另外，使用这类方法需要专业知识，步骤繁琐，而且大量依赖于手工分析证明操作。因此，它们不如模型检测方法实用。

4. 定理证明方法

定理证明方法试图将密码协议的安全性问题当做定理来证明。与程序的正确性证明过程类似，定理证明方法试图将密码协议

的安全性证明规约到一些循环不变式。定理证明方法既可以手工进行，也可以用计算机进行，但其过程比较复杂，是一个新兴的研究热点。在这一领域，Kemmerer、Dutertre、Stephen 等学者做了大量的工作，设计了一些原型系统。但总的来说，这类方法尚不成熟，还需要进一步完善。

5.2 密钥的管理

在上一节所介绍的关于密码协议的安全性分析的假设中，密码系统的安全性取决于用户对密钥的保护。在这一前提下，即使密码体系和算法本身被公开、密码设备丢失，同一类型的密码机制仍可继续使用。然而密钥一旦丢失或出现错误，不仅合法用户不能提取信息，而且还会导致非法的攻击者窃取到机密信息，进而危害到整个系统的安全。由此可见，密钥的管理在通信系统的安全中是极其重要的，它不仅影响着系统的安全性，而且还将涉及系统的可靠性、有效性和经济性等内容。

密钥管理是一个复杂的系统工程，它包括密钥的产生、存储、装入、共享、分配、保护、销毁、证书、保密以及丢失后处理等众多环节。其中，密钥的产生是最基本的问题之一，而密钥的共享、分配和存储则可能是最为棘手的问题。在现实世界中，密钥管理将会不可避免地遇到人事、规章、物理条件等一系列问题。但在本节中，将只限于从理论上讨论作者在密钥的产生、共享、分配等环节中的研究成果。

5.2.1 用户基本密钥的生成

根据密钥的功能和生存域的不同，可以将密钥初步分成基本

密钥、会话密钥、辅助密钥、终端主密钥和主机主密钥等类型。其中，基本密钥又称为初始密钥或用户密钥，是由用户选定或由系统分配给用户专用的，其生存期较长，对安全性要求较高。在本小节中，主要讨论椭圆曲线公钥密码体系下与用户基本密钥相关的一些问题。

在椭圆曲线公钥密码体系中，每个用户拥有一对密钥，其中一个是私有密钥，它仅能为用户私人所拥有，不能公开；另一个被称为公开密钥，是由用户的私钥启动某一算法后产生的，并且可以被公开，供其他需要的用户查阅。

对于本章所要讨论的系列新型椭圆曲线密码体系，通信的双方需事先按照算法5.1产生自己的私钥SK和公钥PK，并将所产生的公钥PK置于可信的第三方认证中心CA上。

对于定义在有限域$GF(q)$上的一条安全椭圆曲线$E(GF(q))$，设在其上随机选取的基点为G，$n=\#E(GF(q))$是椭圆曲线E的阶，r是n的一个大素数因子，则E,q,r,G共同构成了椭圆曲线密码体系的基本参数，这些基本参数一旦确定，如果没有特别的原因，一般是不能变动的。

算法5.1　密钥对选取算法

输入：椭圆曲线E，有限域$GF(q)$，r，基点G 输出：私钥SK，公钥PK
1. 随机选取一整数$s\in[1,r-1]$； 2. 计算$Q=s^{-1}G$，显然Q是椭圆曲线E上的一个点； 3. 私钥$SK=s$； 4. 公钥$PK=Q$。

在上述密钥生成方案中，通信各方所选取的密钥对之间有如下关系：

$$PK \times SK = G \tag{5.1}$$

在利用算法 5.1 生成用户的基本密钥对时，需要注意下列一些问题。

1. 私有密钥的生成

对椭圆曲线密码体系而言，私有密钥既可以由用户自行选定，也可以让计算机随机挑选。在这两种方案中，由用户自行选定的密钥看起来似乎很不错，但因一般的用户会选择容易记忆的词汇或短语作为私钥，故导致所选定的密钥的熵值一般很低，而熵值较差的私钥会严重危害系统的安全。因而在一般情况下，私有密钥需要由计算机随机挑选产生。另一方面，由于受计算机和算法的固有特征限制，任何企图通过计算机算法来获得真正的随机数是不可能的。计算机系统一般会依据一定的伪随机数生成算法来挑选私有密钥，这就使得所生成的密钥难逃被预测的危险。

为了生成一个随机性很好、不可预测的、具有足够大的熵值的随机密钥，通常需要采用诸如掷硬币、扔骰子等随机方式，或者利用离子辐射脉冲检测器、晶体放电管、噪声二极管振荡器和带泄漏的电容等物理噪声信号发生器等工具来获得，以保证所生成的私有密钥的随机性。特别是当用户对所生成的密钥要求较高时，可以采用上述这些方法。

在具有相当规模的网络条件下，上述方法并不可行。幸好，Intel 公司即将在其新的 64 位 Itanium CPU 中内置一种基于电子噪声状态和电流噪声信号等系统热噪音的、输出可变和不可预测的真随机数发生器，可以有效地解决这一问题。

但为了使私有密钥生成系统具有普遍适用性，需要一个能快速产生随机性很好、不可预测的、具有足够大的熵值的、方便实用的真随机密钥生成方法。经过研究，作者提出了自己发明的真随机

密钥生成方法 XRNGS（Xiao's Random Number Generate Scheme）。该方法结合了现有各种密钥产生方法的长处，避免了使用复杂的随机噪声信号发生振荡器以及手工操作的繁琐过程。它根据定点设备随机移动时的方位、速度和指点频率等参量的随机变化情况，获得具有较高的熵值和高度的随机性、难以预测的、高强度的、可靠安全的密钥。该方法能够取代现有的各种密钥产生方法，具有很好的实用价值，适用于所有类型的定点输入设备和各种复杂的应用环境。

针对没有定点输入设备，而又需要能自动地快速产生随机性很好、不可预测的、具有足够大的熵值的真随机密钥的场合，作者也提出和设计了两种基于自然界天然白噪声信号的真随机密钥生成方法及装置。

有关这三种真随机密钥生成技术的详细情况，请参见第7.1节。

2. 公钥的可信度

在公钥密码体系中，公钥可以被公开，以供其他需要的用户查阅。在网络安全假设模型中，网络是被攻击者所控制的。于是，如何判定所查阅的公钥确实是人们所需要的真正的公钥就成了一个大问题，这个问题就是公钥的可信度问题。

在本书所讨论的新型椭圆曲线密码体系中，将用户公钥的真实性和正确性交由可信的第三方认证中心CA来保证。认证中心在接收到用户的公钥后，首先通过检测该公钥是否在椭圆曲线E上来判断该公钥是否有问题，然后，认证中心CA将用户的身份信息和用户的公钥绑定在一起，最后对其进行数字签名制成公钥证书。由于认证中心CA是可信的，所以，可以假定攻击者企图攻入可信认证中心CA，并不可能对包含有合法公钥的证书进行篡改。

而在公钥证书的传递过程中，由于有身份信息和数字签名的双重保护，篡改和伪造公钥数字证书也是不可能的。所以，当通信的接收者通过对公钥证书的签名和身份信息进行检查后，即可判定所收到的公钥证书的真实性和正确性。也就是说，新型椭圆曲线密码体系中公钥的可信度等同于第三方认证中心CA 的可信度。

5.2.2 密钥协商方案

1949 年，Claude Shannon(香农)在“保密系统的信息理论”的论文中证明了一次一密系统的安全性。而经常性地更换密钥过于繁琐，因此在实际操作中是不可行的。为了使得通信的双方能够在不必频繁更换其基本密钥的情况下得到类似于一次一密系统的安全性，人们提出了“会话密钥”的概念。所谓会话密钥，就是指通信的双方在一次通话或数据交换中所使用的临时密钥。

密钥协商方案(Key Agreement Scheme)就是一种能够让通信的双方或者多个参与方在一个公开的、不安全的信道上通过通信协商联合建立一次会话所用的临时密钥的密码协议。在一个密钥协商方案中，若双方能够正确地执行完协议，则最后协商出来的临时会话密钥的值将是一个由参与各方提供的输入共同作用得到的函数值，它对于参与各方而言是相同的。

在椭圆曲线密码体系中，常用的密钥协商方案有 ECKAS-DH (Elliptic Curve Key Agreement Scheme-Diffie-Hellman Version) 方案和 ECKAS-MQV (Elliptic Curve Key Agreement Scheme-Menese-Qu-Vanstone Version) 方案两类，它们都被列入 IEEE 1363 -2000和ANSI-F. 9. 62 等标准之中。

这两种密钥协商方案内容如下。

1. ECKAS-DH 密钥协商方案

设需要完成密钥协商协议的通信双方分别为 A 和 B，则 ECKAS-DH 密钥协商方案可描述为：

① A 随机地选取一个大整数 $k_A \in [1, r-1]$，并计算

$$Q_A = k_A G$$

将其发送给 B。

② B 随机地选取一个大整数 $k_B \in [1, r-1]$，并计算

$$Q_B = k_B G$$

将其发送给 A。

③ A 在收到 B 发来的 Q_B 后，计算

$$K_A = k_A Q_B = k_A k_B G$$

类似地，B 在收到 A 发过来的 Q_A 后，计算

$$K_B = k_B Q_A = k_B k_A G = k_A k_B G$$

这样，A 和 B 完成了 ECKAS-DH 密钥协商方案，协商后双方约定的临时会话密钥为 $K = k_A k_B G$。

ECKAS-DH 密钥协商方案的过程如图 5-2 所示。

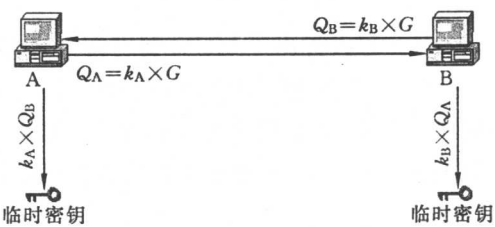

图 5-2　ECKAS-DH 密钥协商方案

ECKAS-DH 密钥协商协议非常简单，它基于椭圆曲线离散对数问题，能够较好地防止被动攻击。但不幸的是，它无法应付一种

所谓的中间人(Man in the Middle)的主动攻击。中间人攻击模式如图 5-3 所示。

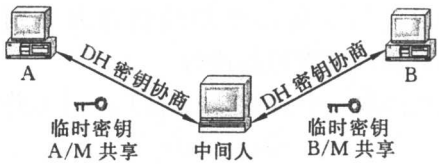

图 5-3 中间人攻击

这样,在协议完成后,A、B 实际上是与中间人侵者 M 完成了密钥协商。A、B 间的通信则完全暴露在 M 面前。

为了解决这一问题,必须确保是用户 A 和 B 正在交换消息而不是与中间人侵者 M 交换消息。要做到这一点,一般的技术是在交换密钥之前以及在协商密钥的同时,通过数字签名和身份鉴别技术,在第三方可信认证中心CA 和密钥分配中心KDC 的帮助下,互相鉴别对方的身份,完成密钥交换过程。这种经过修改的DH 类协议又被称为认证密钥协议。

2. ECKAS-MQV 密钥协商方案

1998 年,Menese 等人提出了 MQV 型密钥共享方案,该方案通过使用双重公钥,可以防止针对上述 DH 型密钥协商方案的中间人攻击。在这一方案中,需要交换密钥的各方实际上都将拥有两对密钥,一对是静态的,另一对是在密钥协商方案的实施过程中动态产生的临时密钥。

设 A 随机选取的私有密钥为整数 $SK_A \in [1, r-1]$,公钥 $PK_A = SK_A \times G$ 置于可信认证中心,(PK_A, SK_A)构成了 A 的静态密钥对。类似地,B 也随机选取的私有密钥 $SK_B \in [1, r-1]$,并将公钥 $PK_B = SK_B \times G$ 置于可信认证中心,B 的静态密钥对是 $(PK_B,$

SK_B)。

具体方案如下。

① A 随机选取整数 $k_A \in [1, r-1]$,计算

$$S_A = k_A G$$

将其发送给B。

② B 随机选取整数 $k_B \in [1, r-1]$,计算

$$S_B = k_B G$$

将其发送给A。

③ A 对于从B 处收到的 S_B,用自己的私钥 SK_A 先计算

$$R_A = (k_A + \overline{S}_A \times SK_A) \bmod r$$

然后利用从认证中心获得的B 的公钥 PK_B 计算

$$K_{AB} = R_A(S_B + \overline{S}_B \times PK_B)$$

④ B 在收到A 发出的 S_A 后,先用自己的私钥 SK_B 计算

$$R_B = (k_B + \overline{S}_B \times SK_B) \bmod r$$

然后利用从认证中心获得的A 的公钥 PK_A 计算

$$K_{BA} = R_B(S_A + \overline{S}_A \times PK_A)$$

ECKAS-MQV 密钥协商方案的正确性证明如下。

因为

$$\begin{aligned} K_{AB} &= R_A(S_B + \overline{S}_B \times PK_B) \\ &= R_A(k_B + \overline{S}_B \times SK_B)G \\ &= R_A R_B G \end{aligned}$$

以及

$$\begin{aligned} K_{BA} &= R_B(S_A + \overline{S}_A \times PK_A) \\ &= R_B(k_A + \overline{S}_A \times SK_A)G \\ &= R_B R_A G \\ &= R_A R_B G \end{aligned}$$

所以，有

$$K = K_{AB} = K_{BA} = R_A R_B G$$

这样，A 和B 完成了ECKAS-MQV 密钥协商方案，协商后双方约定的临时会话密钥为$K=R_A R_B G$。

在上述方案中，$\overline{R}$ 为一种从椭圆曲线群上的点到有限域$GF(q)$中元素的一个单射。例如，设$\overline{R}$ 为取椭圆曲线中点R 的x 分量。

ECKAS-MQV 密钥协商方案较好地解决了DH 协议所存在的中间人主动攻击问题，又避免了因采用数字签名和身份鉴别技术，以及因CA 和KDC 的加入而带来的额外的时间、空间和网络通信开销，以及由于繁琐的认证步骤而可能带来的安全隐患，但其计算步骤仍略显复杂，计算量较大。

作者在分析了上述两类典型的密钥协商方案和其他一些密钥协商方案的优缺点之后，基于DH 类方案和MQV 类方案的思想，提出了改进，即 XKAS 密钥协商方案(Xiao's Key Agreement Scheme)。

3. XKAS 密钥协商方案

在XKAS 方案中，有如下约定。

设椭圆曲线$E(GF(q))$是定义在有限域$GF(q)$上的一条安全椭圆曲线，在其上随机选取的基点为G，设$n=\#E(GF(q))$是椭圆曲线E 的阶，r 是n 的大素数因子。则通信的双方A 和B 须事先按照算法5.1 产生自己的私钥SK 和公钥PK，并将所产生的公钥PK 置于可信的第三方认证中心CA 上。

当通信的双方需要通过不安全的信道协商本次通信所用的临时会话密钥时，可按如下操作进行密钥协商。

① A 随机选择一整数 $k_A \in [1, r-1]$，并从认证中心 CA 处获取 B 的公钥 PK_B，然后计算 $S_A = k_A PK_B$，并将 S_A 发送给 B；

② B 随机选择一整数 $k_B \in [1, r-1]$，并从认证中心 CA 处获取 A 的公钥 PK_A，然后计算 $S_B = k_B PK_A$，并将 S_B 发送给 A；

③ A 对于从 B 处收到的 S_B，用自己的私钥 SK_A 计算

$$K_{AB} = k_A SK_A S_B$$

类似地，B 在收到 A 发出的 S_A 后，用自己的私钥 SK_B 计算

$$K_{BA} = k_B SK_B S_A$$

下面，给出 XKAS 密钥协商方案的正确性证明。

显然，有

$$\begin{aligned} K_{AB} &= k_A SK_A S_B \\ &= k_A SK_A (k_B PK_A) \\ &= k_A k_B (SK_A \times PK_A) \\ &= k_A k_B G \end{aligned}$$

以及

$$\begin{aligned} K_{BA} &= k_B SK_B S_A \\ &= k_B SK_B (k_A PK_B) \\ &= k_B k_A (SK_B \times PK_B) \\ &= k_A k_B G \end{aligned}$$

所以

$$K = K_{AB} = K_{BA} = k_A k_B G$$

至此，通信的双方完成了密钥协商，双方协商后约定的本次通信所用的临时会话密钥为 $K = k_A k_B G$。

需要注意的是，这里所生成的临时密钥是椭圆曲线上的一个点 (K_x, K_y)，不能直接用于对称密码体系的加密和解密操作，如果需要，必须做一些附加操作。这里举例说明这种操作过程。

选用密钥长度为160位的AES-2对称密码算法，为了与之匹配，需要使有限域$GF(p)$的阶达到160位以上。这样，经过上述XKAS密钥协商过程所得到的临时密钥$K=k_Ak_BG=(K_x,K_y)$，将两个坐标上的分量作为串连接起来，可以得到一个320位的串$(K_x \| K_y)$，若不足320位则通过补0，使得串的长度为320位。然后，选用安全杂凑算法SHA-1对所得到的串$(K_x \| K_y)$进行杂凑操作，可获得一个160位杂凑值，该值即为160位的AES对称密码算法的密钥。当选用其他的密钥长度或密码算法时，方法类似。

XKAS密钥协商方案的过程如图5-4所示。

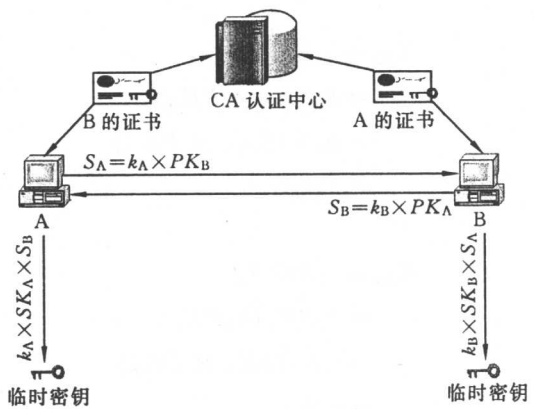

图5-4　XKAS密钥协商方案

(1) XKAS密钥协商方案的性能分析

在XKAS密钥协商方案中，双方各自仅需要进行两次数乘运算和一次通信即可完成密钥交换，而且双方既不需要额外的数字签名和身份鉴别过程，也不需要KDC的协助，避免了额外的通信开销，故优于上面介绍的ECKAS-DH和ECKAS-MQV等密钥协商方案。

(2) XKAS 密钥协商方案的安全性分析

这里使用目前比较成熟的攻击检验方法和形式分析方法来对XKAS密钥协商方案进行安全性分析。

① 攻击检验方法。假设攻击者能够在A和B协商密钥时获取会话密钥,则有两种途径。一种是从通信过程中传递的$S_A=k_APK_B$和$S_B=k_BPK_A$中求出k_A,k_B,进而获取本次通信的临时会话密钥$K=k_Ak_BG$。另一种是希望能够利用式(5.1)从公钥PK获取私钥SK,现设$Q=PK$,$s=SK$,则攻击者必须求解方程

$$sQ=G \tag{5.2}$$

其中,基点G是已知的。若攻击者能够利用Q和G求出s,则能够破解本密钥协商方案。但显然方程(5.2)是一个椭圆曲线离散对数方程。由前文可知,目前尚未出现有效的求解方法,所以,攻击者无法在有限的资源下求出私钥。这样,本方案中的密钥对是安全的,能够很好地抵抗被动攻击。

对于主动攻击,采用前面的密码协议安全性分析假设,认为主动攻击者拥有足够的资源,有能力控制和改变网络上的通信双方之间相互发送的通信内容,这时,主动攻击过程如图5-5所示。

然而,由于攻击者并不知道通信双方的私有密钥SK_A和SK_B,即使篡改了S_A或S_B,也不能伪装成某一方与另一方完成密钥协商过程。也就是说,通信的双方能够确定对方是正确的通信对象。所以,该方案能够较好地抵抗主动攻击。

具体地说,在XKAS方案中,由于使用了公钥证书,通信的双方能够确定对方是唯一的正确的通信对象,没有其他人能够假冒,所以能够很好地抵抗中间人攻击,即XKAS方案也隐藏着身份认证功能。

对重放攻击而言,与其他密钥协商方案相比,XKAS方案虽然没有使用时间戳,但由于采用了随机数,所以,保证了信息的"新

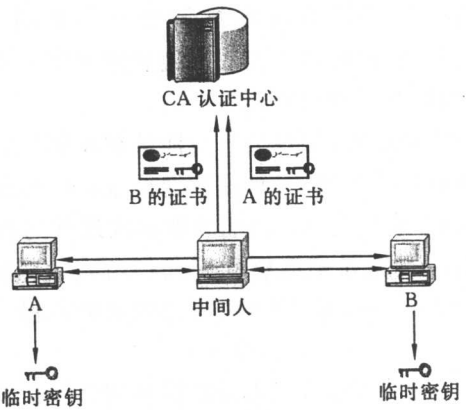

图 5-5　主动攻击 XKAS 密钥协商方案

鲜性”，能够抵抗重放攻击。也就是说，在该方案中，随机数的使用，隐藏了时间性要求。而且由于XKAS 方案完全不依赖时间同步协议，因而也避免了针对时间戳的攻击。

此外，由于在 XKAS 方案中不存在可验证的明文，所以该方案还可以有效地抵抗字典攻击。

② 形式分析方法。这里用目前比较成熟的BAN 类逻辑来说明XKAS 密钥协商方案的安全性。由认证中心CA 的可信度可知

$$A|\equiv|\xrightarrow{K_B}B$$

$$B|\equiv|\xrightarrow{K_A}A$$

由XKAS 密钥协商方案执行步骤可知

$$A|\sim\{N_A\}_{K_B}$$

$$B\lhd N_AG$$

$$A|\equiv B\lhd N_AG$$

$$A|\equiv B\lhd N_AN_BG$$

所以

$$A| \equiv A \xleftrightarrow{K_{AB}} B$$

$$B| \equiv A| \equiv A \xleftrightarrow{K_{AB}} B$$

另一方面

$$B| \sim \{N_B\}_{K_A}$$

$$A \lhd N_B G$$

$$B| \equiv A \lhd N_B G$$

$$B| \equiv A \lhd N_A N_B G$$

因此

$$B| \equiv A \xleftrightarrow{K_{AB}} B$$

$$A| \equiv B| \equiv A \xleftrightarrow{K_{AB}} B$$

所以，XKAS 密钥协商方案是安全的。

同样地，也可以用SMV 转态模型检测工具对XKAS 密钥协商方案进行安全性分析，以说明其安全性。

4. XKAS 密钥交换方案的扩展

(1) 单方XKAS 密钥协商方案

在某些情况下，需要一种特殊的认证，即通信的一方(设为B)希望保持匿名，同时希望能确认另一方的身份，并要求完成密钥协商，保证本次通信的安全。注意，这与一般的密钥分配协议是不同的，在密钥分配协议中，通信双方的身份都是确定的，而在这里，只有一方的身份是确定的。

下面，给出这种情况下的单方XKAS 密钥协商方案。

设通信的双方为A 和B，其中B 为匿名方。设A 事先按照算法5.1 产生的私钥为SK_A，公钥为PK_A，则可按照如下规则协商一次

通信的临时会话密钥。

① B 随机选择一整数$k\in[1,r-1]$，并从认证中心CA 处获取A 的公钥PK_A，然后计算$S=kPK_A$，并将S 发送给A，计算$K=kG$；

② A 对于从B 处收到的S，用自己的私钥SK_A计算$K=SK_A\times S$。

这样，完成了单方密钥协商过程，并约定本次通信所用的临时密钥K。单方XKAS 密钥协商方案如图5-6 所示。

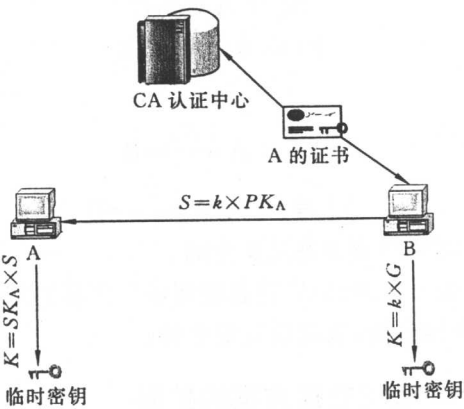

图5-6　单方XKAS 密钥协商方案

(2) 三方XKAS 密钥协商方案

上面研究的XKAS 方案只能应用于双方通信时的密钥协商，但经过适当改进后也可以应用于三方通信时的密钥协商。具体方案如下。

设通信的三方为A、B 和C，他们的密钥对均由算法5.1 产生。设他们的私钥分别为SK_A、SK_B和SK_C，公钥分别为PK_A、PK_B和PK_C，则可按照如下规则协商一次通信的临时会话密钥。

① A 随机选择一整数$k_A\in[1,r-1]$，并从认证中心CA 处获

取C的公钥PK_C，然后计算$X_A=k_APK_C$，并将X_A发送给B；

② B随机选择一整数$k_B\in[1,r-1]$，并从认证中心CA处获取A的公钥PK_A，然后计算$Y_B=k_BPK_A$，并将Y_B发送给C；

③ C随机选择一整数$k_C\in[1,r-1]$，并从认证中心CA处获取B的公钥PK_B，然后计算$Z_C=k_CPK_B$，并将Z_C发送给A；

④ B对于从A处收到的X_A，用刚才所选取的随机整数k_B计算$X_B=k_BX_A$，并将X_B发送给C；

⑤ C对于从B处收到的Y_B，用刚才所选取的随机整数k_C计算$Y_C=k_CY_B$，并将Y_C发送给A；

⑥ A对于从C处收到的Z_C，用刚才所选取的随机整数k_A计算$Z_A=k_AZ_C$，并将Z_A发送给B；

⑦ C对于从B处收到的X_B，用自己的私钥SK_C计算$K_{ABC}=k_CSK_CX_B$；

⑧ A对于从C处收到的Y_C，用自己的私钥SK_A计算$K_{BCA}=k_ASK_AY_C$；

⑨ B对于从A处收到的Z_A，用自己的私钥SK_B计算$K_{CAB}=k_BSK_BZ_A$。

显然，有

$$K=K_{ABC}=K_{BCA}=K_{CAB}=k_Ak_Bk_CG$$

这样就完成了通信三方的密钥协商过程，协商后约定的本次通信所用的临时密钥$K=k_Ak_Bk_CG$。

三方XKAS密钥协商方案如图5-7所示。

由图5-7可知，这一协商方案由三个密钥传递环组成：从A到B到C，从B到C到A和从C到A到B。

对于四方或更多的通信方，只需在上述的三方XKAS密钥协商方案中增加更多的密钥传递环即可。

显然，当通信方很多的时候，该方案的性能很差。这时，建议采

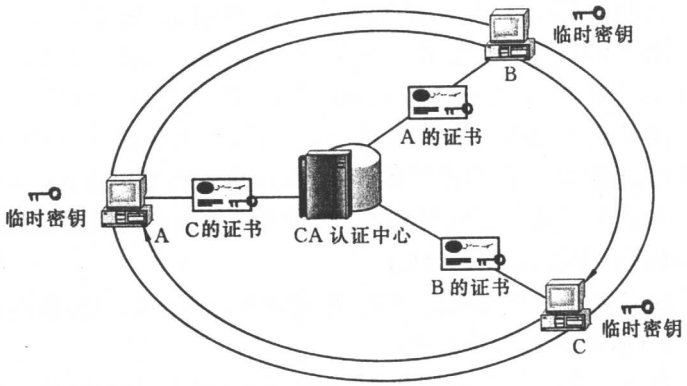

图 5-7　三方 XKAS 密钥协商方案

用下节所要介绍的会议主席制的会议密钥分配方案。

5.2.3　XKDS 密钥分配方案

密钥分配(Key Distribution)是这样一种机制:通信的一方先选择一个秘密的会话密钥,然后将它传输给通信的另一参与方。传统的方法是通过邮递、信使或者专用保密信道传输密钥。密钥可用穿孔纸带或数字形式记录。这类方法一般可以保证密钥传输的及时性,而其安全性则完全取决于信使的忠诚度和素质或者专用信道的安全性,因而使用和维护的成本极高。

设有n个用户参与网络通信,则这类方法需要C_n^2条安全信道。当参与网络通信的用户数n稍大时,用于密钥传递的安全信道数目以及其上的传输量和存储量都非常大,通信成本相当昂贵。

目前常用的密钥分配协议有 Blom 方案、Diffie-Hellman 密钥预分配方案和Kerboros 方案三种。其中,Blom 方案具有可证明的安全性,但其会话密钥是固定的,而且需要专用的安全保密信道;

Diffie-Hellman 方案具有计算安全性，但其会话密钥也是固定的，具有一定的危险性；Kerboros 方案能够在线为每个用户分配新的临时会话密钥，但其需要完全的同步时钟，以确保临时会话密钥的新鲜性(Key Freshness)，而这在实际中是非常困难的。

本节将介绍作者提出的一个基于Diffie-Hellman 密钥预分配方案思想的、能够在线为每个用户分配新的临时会话密钥、具有计算安全性的新型密钥分配方案 XKDS(Xiao's Key Distribution Scheme)。

1. 普通XKDS 密钥分配方案

设通信的双方为A 和B，他们的密钥对均根据算法 5.1 产生。设他们的私钥分别为SK_A和SK_B，公钥分别为PK_A和PK_B，且由A 负责产生临时会话密钥，并向通信方B 分配该临时会话密钥，则点对点型XKDS 密钥分配方案的操作步骤如下。

① A 随机选择一整数$k\in[1,r-1]$，则$K=kG$ 为本次通信的临时会话密钥；

② A 从认证中心CA 处获取B 的公钥PK_B，计算$R=k\times PK_B$，然后用自己的私有密钥对R 进行数字签名，可得

$$S = Sig_A(R)$$

并将S 发送给B；

③ B 对于从A 处收到的S，首先从S 中析出R，然后用从认证中心CA 处获取A 的公钥PK_A和数字签名验证算法对R 的真实性和完整性进行验证，最后再用自己的私钥SK_B计算

$$K = SK_B \times R$$

下面来证明点对点型密钥分配方案的正确性。

由于

$$K = SK_B \times R$$

$$= SK_B \times k \times PK_B$$
$$= kG$$

所以，XKDS 方案正确地完成了密钥分配的任务。类似地，容易证明XKDS 密钥分配方案是一个计算上安全的密钥分配方案。

2. 会议主席制 XKDS 密钥分配方案

上面介绍的是只有两个通信方的密钥分配情况，当通信方不止两个时，这时的密钥分配机制将是：由其中的一个通信方产生密钥，并通过密钥分配协议将所产生的这个临时会话密钥分发给其余的通信方，使得所有的成员共享本次会议的临时密钥。这时，称负责产生和分发密钥的通信主体为会议主席，所用的密钥分配协议为会议主席制的会议密钥分配方案。

这类方案有两种分发模式：单播模式和多播模式。

单播模式实际上是由多个点对点密钥分配模式复合而成的，结合上面所介绍的点对点型 XKDS 密钥分配方案，有如下的单播模式的会议 XKDS 密钥分配方案。

设会议主席为 A，会议的各参与会员为 $M_1, M_2, \cdots, M_n$，记为 M_i。他们的密钥对均根据算法 5.1 产生。现设会议主席的密钥对为 (PK_A, SK_A)，各会员的密钥对为 (PK_i, SK_i)，则工作过程如下。

① 会议主席 A 随机选择一整数 $k \in [1, r-1]$，则本次会议通信的临时会话密钥为 $K = kG$；

② 对于会议的各参与方 $M_1, M_2, \cdots, M_n$，会议主席依次执行点对点型 XKDS 密钥分配方案，向所有其他会员分发 $S_i = Sig_A(k \times PK_i)$；

③ 会员 M_i 在收到会议主席 A 发来的 S_i 后，验证 A 的数字签名，确保 S_i 的真实性和完整性，并用自己的私钥 SK_i 计算

$$K = SK_i \times R_i$$

即可得出本次会议通信的临时会话密钥$K=kG$。

上述单播模式的会议XKDS只适用于中小型会议。当会议的参加人数很多时，该方案的通信开销和计算开销都很大。这时，就需要使用多播模式下的会议密钥分配方案。

多播模式下的会议密钥分配方案是一个刚刚新兴的研究热点，有关该领域的详细内容超出了本书的研究范畴。在RFC2627中对此做了详细的讨论，介绍了一系列的多播密钥管理机制的基本框架。总的思路是建立一个分级树状管理结构，然后使用分级管理策略完成会议密钥的动态管理。

5.3 数据加密

数据加密是密码体系的基本任务之一，自1986年Victer Miller和Neal Konlitz分别独立提出以椭圆曲线离散对数问题作为陷门单向函数的椭圆曲线公钥密码体系的思想以来，许多学者在这一领域做了大量的工作。

关于椭圆曲线公开密钥数据加密有两种类型的编码算法。一类是对明文消息m经过某种分组和编码操作，将其嵌入到椭圆曲线上，使其成为椭圆曲线有限群中的某个点，从而实现基于椭圆曲线的数据加密；另一类算法是将解开单向函数的限门变换成椭圆曲线有限群中的一个点，然后利用该点作为“遮掩”，结合私有解密密钥即可去掉遮掩，解开加密。这类方法中，加密编码变换的结果不是椭圆曲线上的点。

常用的基于椭圆曲线离散对数问题的椭圆曲线公钥数据加密编码算法有Menezes-Vanstone椭圆曲线公开密钥加密算法和ElGamal类离散对数公钥加密算法等。

在本书中，作者所设计的基本密钥对生成算法 5.1 与普通的椭圆曲线基本密钥对生成算法不同。由算法 5.1 生成的密钥在完成密钥共享协议时，具有一些额外的优异性能。为了利用这一优势，作者对上述两种常用公钥加密算法做了适当的修改，得到下面两种加密算法。

1. MVX 椭圆曲线公钥加密算法(ECES-Menezes-Vanstone with Xiao's Extend)

设椭圆曲线 $E(GF(q))$是定义在有限域$GF(q)$上的一条安全椭圆曲线，在其上随机选取的基点为G，设$n=\#E(GF(q))$是椭圆曲线E的阶，r是n的大数因子；设通信主体 A 的基本密钥对是根据算法5.1 产生的，其私钥为SK，公钥PK被置于可信的第三方认证中心CA 上，则MVX 椭圆曲线公钥加密算法如下。

(1) *加密过程*

当通信实体B 需要给A 发送消息M时，实体B 按下列步骤对消息M加密。

① 从可信认证中心CA 上获取A 的公钥PK；

② 将所要加密的明文M分组表示为$GF(q)$上的两个元素(m_1,m_2)；

③ 随机选择一整数$k\in[1,r-1]$，计算$c_0=k\times PK$，$Q=k\times G=(y_1,y_2)$；

④ 定义加密计算式

$$E(m_1,m_2,k)=(c_0,y_1m_1,y_2m_2)=(c_0,c_1,c_2)$$

由加密计算式在有限域$GF(q)$上计算可得密文$c=(c_0,c_1,c_2)$；

⑤ 传输密文$c=(c_0,c_1,c_2)$给A。

(2) *解密过程*

当通信实体A 收到B 发来的密文$c=(c_0,c_1,c_2)$时，实体A 按下

列步骤对密文解密。

① 使用自己的私钥SK，计算$Q=SK\times c_0=k\times G$，得(y_1,y_2)；

② 定义解密计算式

$$D(c_0,c_1,c_2)=(c_0,c_1^{-1}y_1,c_2^{-1}y_2)=(m_1,m_2)$$

在有限域$GF(q)$上计算解密计算式，可得数据$m=(m_1,m_2)$；

③ 将数据$m=(m_1,m_2)$转换、合并成明文M。

使用MVX算法能够较好地完成数据加密的任务，其安全性基于椭圆曲线离散对数问题的求解困难性。但由于其加密过程中需要完成两次大数乘法运算，而在解密过程中还需要两次求逆运算，所以其加解密速度较慢。

2. X-ElGamal 椭圆曲线公钥加密算法（ECES-ElGamal with Xiao's Extend）

ElGamal型公钥加密算法是一个基于Abel有限群的一种公钥密码体系，它可以很容易地移植到椭圆曲线有限群上，这就是ElGamal型椭圆曲线公钥加密算法。在该算法中，由于没有求逆元素的计算，所以其运行速度优于上面介绍的MVX加密算法。但由于其一次只能对一个由明文M分组转化而成的域元素进行加、解密操作，所以，对于同样长度的密文，其有效信息载荷只有MVX加密算法的一半。作者分析后发现，这是因为ElGamal算法只利用了屏蔽参数$k\times G$的X分量，因此，给出了基于算法5.1的新型基本密钥对下的改进ElGamal算法，即X-ElGamal椭圆曲线公钥加密算法。

设椭圆曲线$E(GF(q))$是定义在有限域$GF(q)$上的一条安全椭圆曲线，在其上随机选取的基点为G，设$n=\#E(GF(q))$是椭圆曲线E的阶，r是n的大数因子；设通信主体A的基本密钥对是根据算法5.1产生的，其私钥为SK，公钥PK被置于可信的第三方认

证中心CA 上，则X-ElGamal 椭圆曲线公钥加密算法如下。

(1) 加密过程

当通信实体B需要给A 发送消息M时，实体B 按下列步骤对消息M加密。

① 从可信认证中心CA 上获取A 的公钥PK；

② 将所要加密的明文M分组表示为$GF(q)$上的一对元素$m=(m_1,m_2)$；

③ 随机选择一整数$k\in[1,r-1]$，计算$c_0=k\times PK$，$Q=k\times G=(y_1,y_2)$；

④ 定义加密计算式

$$E(m_1,m_2,k)=(c_0,y_1+m_1,y_2+m_2)=(c_0,c_1,c_2)$$

在有限域$GF(q)$上计算上述加密计算式，可得密文$c=(c_0,c_1,c_2)$；

⑤ 传输密文$c=(c_0,c_1,c_2)$给A。

(2) 解密过程

当通信实体A 收到B 发来的密文$c=(c_0,c_1,c_2)$时，实体A 按下列步骤对密文解密。

① 使用自己的私钥SK，计算$Q=SK\times c_0=k\times G$，得(y_1,y_2)；

② 定义解密计算式

$$D(c_0,c_1,c_2)=(c_0,c_1-y_1,c_2-y_2)=(m_1,m_2)$$

在有限域$GF(q)$上应用上述解密计算式，计算出数据$m=(m_1,m_2)$；

③ 将数据$m=(m_1,m_2)$转换、合并成明文M。

X-ElGamal 型椭圆曲线公钥加密算法的正确性证明如下。

由数据加密过程可知

$$\begin{cases}c_1=y_1+m_1\\c_2=y_2+m_2\end{cases}$$

而由解密过程

$$\begin{cases} m_1 = c_1 - y_1 \\ m_2 = c_2 - y_2 \end{cases}$$

显然，明文数据m在经过加密过程和解密过程后，能够正确还原为原来的明文数据m，所以X-ElGamal型椭圆曲线公钥加密算法是正确的。

实验表明，在同等条件下，X-ElGamal型椭圆曲线公钥加密算法的运行速度是MVX算法的20～30倍，在Intel Pentium III 700的计算机上，其加密速度可以达到0.008 MB/s。X-ElGamal算法虽然具有较高的加密效率和性能，但其运行性能仍远远低于同等安全性的对称密码加密算法的加密性能，如AES算法的加密速度可达30.325 MB/s。因此，在实际需要进行数据加密操作时，需要使用混合密码加密方案，即用对称密码加密算法完成对消息M的加密编码，然后将对称算法的密钥用公钥加密算法加密。这一思想在实际应用中比较普遍，如常用的PGP加密方案就是RSA算法和IDEA算法的混合密码体系。

类似地，选择一个合适的对称密码算法（如AES算法），结合某一种快速椭圆曲线公钥密码体系（如X-ElGamal算法），即可实现快速加密。

3. XHES椭圆曲线混合加密算法

经过研究发现，可以结合密钥共享体系对上面所介绍的混合密码加密体系的方案做更深入的改进，以求进一步地提高加密体系的性能。其思想如下。

首先选择合适的密钥共享协议完成随机选择的可用于对称密码算法的加密密钥。然后，再利用该加密密钥对消息明文M进行快速加密。

基于上述密钥共享思想的混合加密体系的具体实现方案XHES(Xiao's Hybird Encryption Scheme)的算法工作流程如图5-8所示。

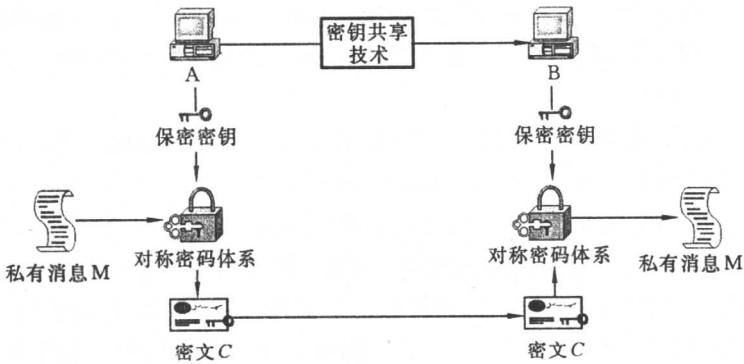

图5-8　XHES椭圆曲线混合加密算法的工作过程

在本算法中，有如下约定。

① 基本密钥对的生成。设椭圆曲线$E(GF(q))$是定义在有限域$GF(q)$上的一条安全椭圆曲线，其上随机选取的基点为G，设$n=\#E(GF(q))$是椭圆曲线E的阶，r是n的大数因子；设通信主体A的基本密钥对是根据算法5.1产生的，其私钥为SK，公钥PK被置于可信的第三方认证中心CA处。

② 对称密码算法的选择。目前常用的对称密码算法有AES、IDEA、RC6、Mars、Serpent等多种。根据目前对称密码算法领域的发展情况，建议选用AES算法来完成实际的数据加密工作。

出于一般化考虑，这里用$Enc(K,M)$来表示所选的对称密码算法的加密过程，用$Dec(K,C)$表示所选的对称密码算法的解密过程。

③ 密钥共享协议的选择。在数据加密的应用条件下，加密密

钥实际上是由一方产生的会话密钥。所以，这里选用单方XKAS密钥协商方案作为密钥共享协议，当然，也可以选用其他密钥共享协议。

④ 加密密钥的变换。由上一节可知，单方XKAS密钥协商方案产生的临时会话密钥是椭圆曲线E上的一个点，所以，还需要根据对称密码算法的需要对该临时会话密钥做进一步的变换，将其转换为适合对称密码算法的加密密钥。这里，设该加密密钥变换算法为H。

基于单方XKAS密钥协商方案的XHES椭圆曲线混合加密算法如下。

(1) 加密过程

当通信实体B需要给A发送消息M时，实体B按下列步骤对消息M加密。

① 从可信认证中心CA处获取A的公钥PK；

② 随机选择一整数$k\in[1,r-1]$，计算$S=k\times PK$和加密密钥$K=H(k\times G)$；

③ 根据第②步所得到的加密密钥，用所选的对称密码算法$Enc(K,M)$对明文M进行加密，设所得到的密文为C；

④ 将数据报文(S,C)传输给A。

(2) 解密过程

当通信实体A收到B发来的数据报文(S,C)时，可按下列步骤对密文解密。

① 使用自己的私钥SK，计算$K=H(SK\times S)=H(k\times G)$，求出加密密钥$K$；

② 根据所求出的加密密钥K，用所选的对称密码算法$Dec(K,C)$对密文C进行解密，即可获得实体B要传输的消息明文M。

4. 带身份鉴别和认证功能的XHES混合加密算法

在实际应用混合密码编码体系时，有时需要附加身份鉴别和认证等功能。这时，可以选择具备身份鉴别和认证功能的点对点型XKDS密钥分配方案作为XHES混合密码加密算法中的密钥共享协议，则此时的XHES混合密码加密方案如下。

(1) 加密过程

当通信实体B需要给A发送消息M时，实体B按下列步骤对消息M加密。

① 从可信认证中心CA处获取A的公钥PK；

② 随机选择一整数$k\in[1,r-1]$，计算$R=k\times PK$和加密密钥$K=H(k\times G)$；

③ 用自己的私钥SK_B对R进行数字签名，得

$$S = Sig_B(R)$$

④ 根据第②步所得到的加密密钥，用所选的对称密码算法$Enc(K,M)$对明文M进行加密，设所得到的密文为C；

⑤ 将数据报文(S,C)传输给A。

(2) 解密过程

当通信实体A收到B发来的数据报文(S,C)时，可按下列步骤对密文解密。

① 从S中析出R，并用从认证中心CA处获取A的公钥PK_A和数字签名验证算法对R的真实性和完整性进行验证；

② 计算$K=H(SK\times R)=H(k\times G)$，求出加密密钥$K$；

③ 使用自己的私钥SK，根据第②步所求出的加密密钥K，用所选的对称密码算法$Dec(K,C)$对密文C进行解密，即可获得实体B要传输的消息明文M。

由XHES混合密码加密算法的工作过程可知，其正确性依赖

于密钥共享协议和对称密码编码算法的正确性，而密钥共享协议和对称密码编码算法的正确性都是已知的，所以XHES混合密码加密算法是正确的。

利用XHES混合密码编码体系，可以以接近对称密码体系的加解密速度完成大量明文消息的加密，同时又具有公钥密码体系的优点。因此，该算法具有很广阔的实用价值。

5.4 数字签名

在现实生活中，人们常常需要进行身份鉴别、数据完整性认证和抗否认。身份鉴别允许我们确认一个人的身份；数据完整性认证则帮助我们识别消息的真伪、是否完整；抗否认则防止人们否认自己曾经做过的行为。传统商业中的契约以及个人之间的书信等常常采用手写签名、印章和封印等手段，以便获得在法律上认可的身份鉴别、数据完整性认证和抗否认效果。而在网络化的数字空间中，为保证电子商务活动的正常进行，人们迫切需要一种能够进行身份认证、来源鉴别、安全保密、数据完整性认证，以及抗抵赖、抗否认的安全协议，这样，数字签名(Digital Signature)应运而生，成为网络安全解决方案的重要基础，推动和加速了网络电子商务活动的发展。

为了实现在网络环境下电子商务交易中的身份鉴别、数据完整性认证和抗否认、抗抵赖等功能，数字签名应满足以下要求。

① 签名者发出签名的消息后，就不能再否认自己所签发的消息；

② 接收者能够确认或证实签名者的签名，但不能否认；

③ 任何人都不能伪造签名；

④ 第三方可以确认收发双方之间的消息传输，但不能伪造这一过程。这样，当通信的双方关于某一消息的数字签名的真伪发生争执时，可以由可信的第三方仲裁来解决双方的争执。

一个典型的数字签名系统的工作过程如图 5-9 所示。

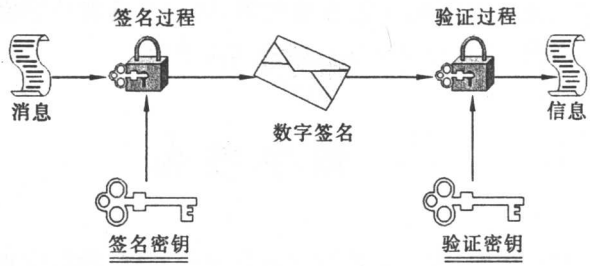

图 5-9　典型的数字签名过程

一个典型的数字签名体系必须包括两个重要的组成部分，即签名算法（Signature Algorithm）和验证算法（Verification Algorithm）。

为了满足上述四项要求，数字签名体系必须满足两条基本假设：

① 签名密钥是安全的，只有其拥有者才能使用；

② 使用签名密钥是产生数字签名的唯一途径。

这样，发送者和接收者必须使用不同的算法过程和操作参数对消息进行签名和验证签名、核实消息的正确性，以防止伪造，保证数字签名的唯一性。

在公钥密码体系中，签名密钥一般就是签名者的私有密钥，而验证密钥则是签名者的公开密钥。

本节将专门介绍基于椭圆曲线公钥密码体系的各种数字签名方案。

5.4.1 XECDS普通数字签名方案

现有的可以被移植到椭圆曲线有限群上的普通数字签名方案(如DSA、ElGamal、Schnorr、Okamoto等)均属于基于离散对数问题的ElGamal类数字签名方案。也就是说,可以将它们规约到某一统一的形式,使得现有的这些普通数字签名方案都成为其特例。

在下面的讨论中,对椭圆曲线密码体系的基本参数有如下约定。

设椭圆曲线$E(GF(q))$是定义在有限域$GF(q)$上的一条安全椭圆曲线,在其上随机选取的基点为G,设$n=\#E(GF(q))$是椭圆曲线E的阶,r是n的大数因子。设通信主体A的基本密钥对是根据算法5.1产生的,其私钥为SK,公钥PK被置于可信的第三方认证中心CA处。

这样,基于椭圆曲线离散对数问题的普通数字签名方案——Xiao's Elliptic Curve Digital Signature(XECDS)可描述如下。

1. 签名过程

设待签名的消息为m,签名值为s,签名者A按如下步骤对m进行签名。

① 计算消息m的杂凑摘要值$H(m)$;

② 随机选择一大整数$k\in[1,r-1]$,计算$Q=k\times G$;

③ 设签名方程为

$$C=(A\times k-B)\times SK \tag{5.3}$$

其中,方程系数A、B和C为$H(m)$,Q或s的函数;

④ 从签名方程中求出s,然后以(Q,s)作为消息m的数字签名。

2. 验证过程

当接收者收到消息 m 和数字签名(Q,s)后，可以按照如下步骤对待核实的消息签名进行验证。

① 计算消息 m 的杂凑摘要值 $H(m)$；

② 设验证方程为

$$A \times Q = B \times G + C \times PK \tag{5.4}$$

式中，方程系数A、B 和C 与签名方程一致，也是为$H(m)$，Q 或s 的函数；

③ 代入 $H(m)$，Q 和 s 的值，若验证方程成立，则签名为真；反之为假。

显然，签名方程式(5.3)和验证方程式(5.4)是等价的，所以XECDS 数字签名方案是正确的。

XECDS 方案的工作过程如图 5-10 所示。

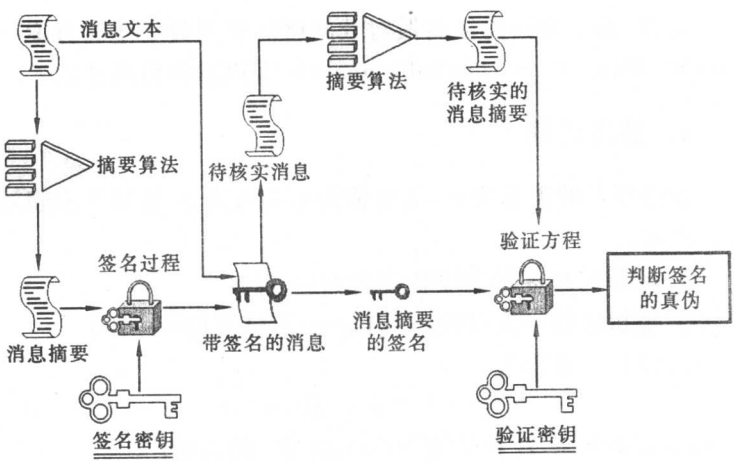

图 5-10　XECDS 方案的工作过程

下面，通过选取一组特定的方程系数，利用上述的XECDS签名方案来构造一个基于椭圆曲线离散对数问题的具体的XECDS-I签名体系。（略加修改，该签名体系还可以转化为一个盲数字签名方案。）

令$A=H(m)$，$B=s$以及$C=-Q_x \bmod r$，则在XECDS签名体系中，由签名方程(5.3)和验证方程(5.4)得

签名方程

$$Q_x=[s-H(m)\times k]\times SK \tag{5.5}$$

验证方程

$$H(m)\times Q=s\times G-Q_x\times PK \tag{5.6}$$

据此，可构成如下XECDS-I签名方案。

（签名过程）设待签名的消息为m，签名者A按如下步骤对m进行签名。

① 用消息摘要算法计算消息m的杂凑摘要值$H(m)$；

② 随机选择一大整数$k\in[1,r-1]$，计算$Q=k\times G$；

③ 由签名方程(5.5)求解s，可得

$$s=H(m)\times k+Q_x\times SK^{-1}$$

则消息m的数字签名为(Q,s)。

（验证过程）设待核实的消息为m，其数字签名为(Q,s)，接收方B可按如下步骤对其进行验证。

① 用消息摘要算法计算消息m的杂凑摘要值$H(m)$；

② 将s，$H(m)$和Q_x代入验证方程式(5.6)，验证其是否成立；

③ 若验证方程式成立，则签名为真；反之为假。

类似地，若令$A=s$，$B=H(m)$以及$C=Q_x \bmod r$，得到的是XEC-DSA签名方案；若令$A=s$，$B=m$以及$C=-Q_x \bmod r$，可以获得XEC-ElGamal签名方案；而令$A=1$，$B=-s$以及$C=H(Q_x,m)$，则能够得到XEC-Schnorr签名方案。

5.4.2 加密与签名

在某些场合下，签名者A希望在向接收方B发送消息时，不仅能够对所发送的消息进行数字签名，而且还希望能够同时对消息进行加密。那么在这种情况下，A需要将加密技术和签名方案结合起来。显然，这时可以有两种结合方式：一种是先对所要发送的消息进行加密，然后对密文进行签名，简称“先加密，后签名”；另一种是先对消息进行数字签名，然后对签过名的消息进行加密，最后将密文发送，简称“先签名，后加密”。

具体地说，设给定的明文消息为M，则这两种方案的工作过程如下。

1. “先加密，后签名”方案

① 发送方A从可信CA中心获取接收方B的公钥PK_B；

② 发送方A利用接收方B的公钥PK_B和相应的公钥加密算法对所要发送的明文M进行加密，所得密文为R，这一过程为

$$R = Enc(PK_B, M)$$

③ 发送方A利用自己的签名密钥SK_A和相应签名算法对密文R进行数字签名，可得

$$S = Sig_A(R)$$

④ 发送方A将(R,S)发送给接收方B；

⑤ 接收方B在接收到报文(R,S)后，先从可信CA中心获取发送者的验证密钥PK_A；

⑥ 接收方B利用发送方A的验证密钥PK_A和相应的签名验证算法对数字签名S的真伪进行验证，即判断方程$Ver(PK_A, R, S)=0$是否成立；

⑦ 若验证方程成立，则签名为真，接收方B可利用自己的私钥SK_B解密密文R，进一步求出明文

$$M = Dec(SK_B, R)$$

仔细分析可以发现，这种方案存在一个潜在的安全隐患，具体说明如下。

设有一个主动攻击者I截获了A发给B的数据报文(R,S)，并企图让接收方B认为该数据报文是攻击者I发送的。他可以这样做：首先利用自己的签名密钥SK_I和相应签名算法对密文R进行数字签名，即$S' = Sig_I(R)$；然后用S'替换S可得新报文(R,S')，然后将伪造的报文(R,S')发送给B；而接收方B在收到伪报文(R,S')后，会利用I的验证密钥PK_I和相应的验证算法，通过判断方程$Ver(PK_I,R,S')=0$是否成立来验证数字签名S'的合法性。结果，导致接收方B会误认为消息M来自攻击者I。

由于这一潜在的安全隐患，不推荐使用“先加密，后签名”方案。

2. “先签名，后加密”方案

① 发送方A首先利用自己的签名密钥SK_A和相应签名算法对消息M进行数字签名，即

$$S = Sig_A(M)$$

② 发送方A从可信CA中心获取接收方B的公钥PK_B；

③ 发送方A利用接收方B的公钥PK_B和相应的公钥加密算法对所要发送的明文数据报文(M,S)进行加密，所得密文为C，这一过程为

$$C = Enc(PK_B, M, S)$$

④ 发送方将密文C发送给接收方B；

⑤ 接收方B在接收到报文C后，利用自己的私钥SK_B解密密

文C，求出带数字签名的明文消息(M,S)；

⑥ 接收方B从可信CA中心获取发送方的验证密钥PK_A；

⑦ 接收方B利用发送方A的验证密钥PK_A和相应的签名验证算法，通过判断方程$Ver(PK_1,R,S')=0$是否成立来验证数字签名S的合法性。若验证方程成立，则签名为真；反之为假。

经分析可知，“先签名，后加密”方案不存在“先加密，后签名”方案所存在的安全隐患，因此，当需要同时使用数字签名和加密技术时，建议使用“先签名，后加密”方案。

5.4.3 盲数字签名方案

一般情况下，人们总是先知道文件的内容，然后再对该文件进行签名。而在某种特殊的情况下，用户需要让签名者对文件进行数字签名，而又不希望签名者知晓文件的具体内容，这就需要盲数字签名(Blind Digital Signature)。

盲数字签名在诸如电子投票选举、电子拍卖、电子支付等需要保护某些参加者的场合中具有广泛而重要的应用。它具有两个重要的特征。

① 签名者能够在不知晓被签名的文件内容的情况下对文件进行签名，即文件的具体内容对于签名者而言是不可见的，或者说是“盲”的。

② 即使签名者看到了被签名的文件以及他对该文件的签名，他也不能判断出这个签名是他在什么时候为什么人签署的，即签名者不能跟踪他所签署的盲签名。

例如：顾客甲向商家乙购买商品，通过中介银行进行电子交易。双方都不希望具体的交易内容被银行得知，但又需要银行对交易过程进行担保，以防诈骗。这时，就需要由银行对交易明细单进

行盲数字签名。

在盲数字签名方案中，称消息的拥有者即需要盲签名服务的通信主体A为用户，而称提供盲签名服务的通信主体B为签名者。

当用户A需要签名者B对消息M进行盲签名时，按下列操作步骤来完成盲签名。

① 用户A首先对等待签名的消息M进行盲变换T，使得消息M的具体内容对于签名者B而言是“盲”的；

② 用户A将变换后的消息m(即盲消息)发送给签名者B；

③ 签名者B对所收到盲消息m进行数字签名，得到签名s；

④ 签名者B将盲消息m及其签名s一起交给用户A；

⑤ 用户A对所收到的签名s做逆盲变换T^{-1}，所得到的就是原消息M的盲签名S。

图5-11所示为上述的普通盲数字签名的签名过程。

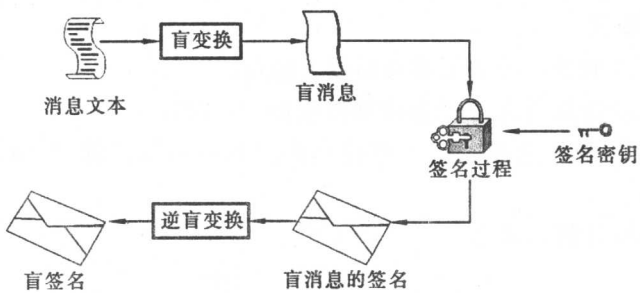

图5-11　普通盲数字签名的签名过程

显然，在上面的盲数字签名的签名过程中，盲变换T使得盲数字签名的“盲”特征得以实现，而逆盲变换T^{-1}则保证了盲签名与原消息是概率无关的，使得签名者事后不能跟踪该盲签名。

下面，给出一个基于第5.4.1节所设计的XECDS-I签名方案的盲数字签名方案XECBDS(Xiao's Elliptic Curve Blind Digital

Signature)。

为了完成对消息 M 的盲签名，用户 A 和签名者 B 执行如下协议。

1. 参数初始化

XECBDS 密码体系的参数同 XECDS-I 签名方案，由签名者 B 完成如下初始化步骤。

① 随机选取一大整数 $k \in [1, r-1]$；

② 计算 $K = k \times G$；

③ 将 K 发送给 A。

2. 盲变换过程

设待签名的消息为 M，经盲变换后的盲消息为 m，用户 A 完成如下步骤。

① 验证 K 是否是椭圆曲线上的点；

② 计算消息 M 的杂凑摘要值 $h = H(M)$；

③ 随机选择一对大整数 $\alpha, \beta \in [1, r-1]$，计算 $R = \alpha \times K + \beta \times G$；

④ 计算盲消息

$$m = \alpha K_x R_x^{-1} \times h$$

式中，K_x, R_x 分别为椭圆曲线上点 K, R 的 x 坐标分量；

⑤ 将盲消息 m 发送给签名者 B。

3. 签名过程

设待签名的盲消息为 m，签名值为 $\tilde{s}$，签名者 B 按如下步骤对 m 进行签名。

① 设签名方程为 $K_x = (\tilde{s} - k \times m) \times SK$；

② 从签名方程中求出 $\tilde{s}=k\times m+K_x\times SK^{-1}$；

③ 将盲消息 m 的签名 $\tilde{s}$ 发送给用户 A。

4. 逆盲变换过程

当接收者收到盲消息 m 的签名 $\tilde{s}$ 后，用户 A 完成如下的逆盲变换过程。

① 计算 $s=\tilde{s}\times R_x\times K_x^{-1}+\beta\times h$；

② 计算 $t=R_x \bmod r$；

③ 以 (t,s) 作为签名者 B 对消息 M 的盲数字签名。

5. 验证过程

当接收者收到消息 M 和盲数字签名 (t,s) 后，按照如下步骤验证盲数字签名的合法性。

① 用消息摘要算法计算消息 M 的杂凑摘要值 $h=H(M)$；

② 计算 $T=h^{-1}\times(s\times G-t\times PK)$；

③ 设 T 的 x 坐标分量为 T_x，验证等式 $t=T_x \bmod r$ 是否成立。若等式成立，则盲数字签名 (t,s) 是由签名者 B 签发的一个有效的盲数字签名；反之该盲数字签名是无效的，即消息文件或盲数字签名在传输过程中被篡改。

上述 XECBDS 盲数字签名方案的正确性证明如下。

由于

$$\begin{aligned}T&=h^{-1}\times(s\times G-t\times PK)\\&=h^{-1}\times(\tilde{s}\times R_x\times K_x^{-1}+\beta\times h-R_x\times SK^{-1})\times G\\&=h^{-1}\times((k\times m+K_x\times SK^{-1})\times R_x\times K_x^{-1}+\beta\times h-\\&\quad R_x\times SK^{-1})\times G\\&=h^{-1}\times(k\times m\times R_x\times K_x^{-1}+SK^{-1}\times R_x+\\&\quad\beta\times h-R_x\times SK^{-1})\times G\end{aligned}$$

$$=h^{-1}\times(k\times m\times R_x\times K_x^{-1}+\beta\times h)\times G$$
$$=h^{-1}\times(k\times\alpha\times h+\beta\times h)\times G$$
$$=(k\times\alpha+\beta)\times G$$
$$=\alpha\times K+\beta\times G$$
$$=R$$

所以，XECBDS 盲数字签名方案是正确的。

对签名者 B 而言，由于 α,β 是由用户 A 完全随机选取的，所以，盲消息 m 对签名者B 而言是不可见的，而且与原消息 M 是概率无关的。

总之，盲签名具有消息内容的保密性以及盲签名与原消息的概率无关等特征，较好地保护了消息通信主体的个人隐私，具有较为广泛的应用。

5.4.4 代理数字签名方案

在现实生活中，为了解决签名权力的可转移性问题，人们发明了印章，使得签名权力与具体的签名盖章人无关。当具体的签名盖章人变动时，检验文件印章真伪的方法并不需要改变。在数字社会中，代理数字签名(Proxy Digital Signature)方案是一种用于解决数字签名权力的委托代理转移问题的数字签名方案，它是由 Mambo. M. 于1996 年首先提出的。其主要功能就是实现类似现实生活中的印章的这种转移签名权力的功能。代理数字签名方案使得用户可以将自己的签名权力委托给另一个可信的代理人，并由该代理人代替他们签发某一消息，所留下的数字签名就是代理数字签名。

在代理数字签名方案中，提出委托的用户被称为原始签名者(Original Signer)，可以将自己的数字签名权力委托给另一个可信

的被称为代理签名者(Proxy Signer)的受托方,使得代理签名者在不知道原始签名者的签名密钥的情况下,代表原始签名者行使实际的签名权力,根据电子文件,生成特定消息的数字签名,即代理数字签名。相应地,称能够生成代理数字签名的数字签名方案为代理数字签名方案(Proxy Digital Signature Scheme)。

1. 代理数字签名方案的性质

为了保证代理数字签名的正常运作,一个代理数字签名方案应该满足方案的基本不可伪造性、代理签名的不可伪造性、代理签名的可区分性、代理签名的不可抵赖性、身份的可证实性、密钥的依赖性等基本性质,具体说明如下。

(1) 基本性质

除了原始签名者外,任何人(包括代理签名者)都不能生成原始签名者的普通数字签名。这条性质被称为基本的不可伪造性,它可以保证原始签名者的基本安全要求,是任何数字签名体系都应当具备的性质。

(2) 不可伪造性

除了代理签名者外,任何人(包括原始签名者)都不能生成有效的代理签名。特别地,如果原始签名者委托了多个代理签名者,那么任何代理签名者都不能伪造其他代理签名者的代理签名。这条性质被称为代理数字签名的不可伪造性,它可以保证代理签名者的基本安全要求。

(3) 可区分性

任何一个代理数字签名都与原始签名者的普通数字签名有明显的区别,而不同代理签名者所生成的代理数字签名之间又可以被明显区分。这条性质被称为代理数字签名的可区分性,和前述的性质结合起来,可以用于防止签名者之间相互抵赖。

(4) 不可抵赖性

任何签名者(不论是原始签名者还是代理签名者)在生成一个数字签名后,不能再对它加以否认。这条性质被称为代理数字签名的不可抵赖性。

(5) 可跟踪性

原始签名者可以根据某一有效的代理数字签名确定是哪一个代理签名者签发了该消息,确定出相应的代理签名者的身份。这条性质被称为代理数字签名的可跟踪性(签名者身份的可证实性)。利用这一性质,原始签名者可以对代理签名者进行事后监督,对所签发的代理数字签名进行跟踪,防止代理签名者滥用其所获得的代理签名权力。

(6) 密钥依赖性

代理签名密钥的产生依赖于原始签名者的私钥,这条性质被称为代理数字签名的密钥依赖性。

(7) 可注销性

如果原始签名者希望代理签名者只能在一定时间区间内拥有代理签名的能力,那么必须能够让代理签名者的代理签名密钥在指定的时刻失去作用。这条性质被称为代理数字签名密钥的可注销性,一般都是通过网络系统广播来实现的。它使得原始签名者能够控制代理签名者的授权期限,使得代理签名者只能在一定的时间范围内拥有签发代理数字签名的能力。

代理数字签名有效地模拟了现实生活中印章的功能,较好地解决了数字签名权利的委托代理问题,具有比较重要的使用价值。

一般而言,一个代理数字签名方案由以下三个核心过程组成。

① 委托过程。在这一过程中,原始签名者将通过某种方式,完成向代理签名者的数字签名权力的委托转移过程。根据委托方式

的不同，这一过程又进一步地被细分为完全委托型(Full Delegation)、部分委托型(Partial Delegation)两种类型的委托方式。其中，完全委托型委托方式因为安全性问题已经不再被使用；部分委托型委托方式使代理签名者能够在不获得原始签名者的主签名密钥的情况下，完成对特定消息的数字签名，且其生成和验证过程所需要的工作量与普通数字签名基本相当，因而成为当前主要的委托方式。

② 签名过程。代理签名者在接收到原始签名者转移的数字签名权力后，代表原始签名者对特定消息完成代理数字签名。

③ 验证过程。接收方在收到代理数字签名后，利用验证算法验证该代理数字签名的合法性。

2. XECPDS 代理数字签名方案

下面，给出一个基于第5.4.1节所设计的XECDS-I签名方案和Mambo. M. 提出的代理签名数字方案的基本的代理数字签名方案XECPDS(Xiao's Elliptic Curve Proxy Digital Signature)，其基本的密码体系参数同XECDS-I签名方案。

为了完成对消息 m 的代理数字签名，原始签名者A和代理签名者B执行如下协议。

(1) 委托过程

① 原始签名者A随机选取整数 $k\in[1,r-1]$，计算 $Q=k\times G$；

② 原始签名者A计算 $SK_{\sigma}^{-1}=SK_{A}^{-1}+k\times Q_x$，并将 SK_{σ} 通过秘密渠道发送给代理签名者B，而 Q 则以公开的方式直接发送给代理签名者B，委托信息即为 (SK_{σ},Q)；

③ 代理签名者B在收到委托信息 (SK_{σ},Q) 之后，首先需要验证等式 $SK_{\sigma}^{-1}\times G=PK_{A}+Q_x\times Q$ 是否成立。若等式不成立，则说明所收到的委托信息无效，应予以重新发送；若等式成立，则说明

所收到的委托信息有效；

④ 代理签名者B利用自己的签名私钥SK_B，计算代理签名私钥$SK_\sigma^{-1}=SK_\sigma^{-1}+SK_B^{-1}\times(PK_B)_x$。

（2）签名过程

对消息m，代理签名者B使用SK_σ作为签名私钥，利用事先约定的普通数字签名协议生成普通数字签名$s=Sig_n(SK_\sigma,m)$，则代理签名者B代表原始签名者A对消息m生成的代理数字签名为(s,Q)。

（3）验证过程

对于所收到的消息m和代理签名(s,Q)，验证步骤如下。

① 计算$PK_\sigma=PK_A+Q_x\times Q+(PK_B)_x\times PK_B$；

② 利用普通数字签名的签名验证方程验证签名的合法性，即

$$Ver(PK_A,(s,Q),m)=\text{True}$$
$$\Leftrightarrow Ver(PK_\sigma,s,m)=\text{True}$$

此时，$PK_\sigma=PK_A+Q_x\times Q+(PK_B)_x\times PK_B$实际上可看成是与代理签名私钥$SK_\sigma$相对应的代理签名公钥。

该代理数字签名方案的正确性证明如下。

$$\begin{aligned}PK_\sigma&=PK_A+Q_x\times Q+(PK_B)_x\times PK_B\\&=SK_A^{-1}\times G+Q_x\times k\times G+(PK_B)_x\times SK_B^{-1}\times G\\&=(SK_A^{-1}+k\times Q_x+(PK_B)_x\times SK_B^{-1})\times G\\&=((SK_A^{-1}+k\times Q_x)+(PK_B)_x\times SK_B^{-1})\times G\\&=(SK_\sigma^{-1}+SK_B^{-1}\times(PK_B)_x)\times G\\&=SK_\sigma^{-1}\times G\\&=PK_\sigma\end{aligned}$$

下面来讨论一下该方案所满足的代理数字签名的基本性质。

① 基本的不可伪造性。由于代理签名者B无法根据所收到的

委托信息(SK_{ω},Q)计算出原始签名者A的签名私钥SK_A，因此不可能伪造原始签名者A的普通数字签名。进而可知其他攻击者都不能生成原始签名者A的普通数字签名。

② 代理数字签名的不可伪造性。由于只有代理签名者B才能利用自己的签名私钥SK_B，计算出代理签名私钥SK_{ω}，进而生成消息m的代理数字签名。所以，除了代理签名者B外，任何人(包括原始签名者A)都不能伪造代理签名者B的有效代理数字签名。特别地，即使原始签名者A委托了多个代理签名者$B_1,B_2,\cdots,B_n$，那么任一代理签名者都不能伪造其他代理签名者的有效代理数字签名。

③ 代理数字签名的可区分性。代理签名者B代表原始签名者A对消息m生成的代理数字签名(s,Q)是由普通数字签名s和椭圆曲线上的某一数点Q两部分组成的。由于代理数字签名比普通数字签名多出一部分，显然，将代理数字签名和普通数字签名区分开来很容易。由于数点Q是按公式$Q=k\times G$随机生成的，因此，不同代理签名者所生成的代理数字签名之间也可以被明显区分。

④ 代理数字签名的不可抵赖性。由于任何人都不能伪造原始签名者A的普通数字签名，所以，A不能否认它的一个有效的普通数字签名。同理，由于除了代理签名者B外，任何人(包括原始签名者A)都不能伪造代理签名者B的有效代理数字签名，B也不能否认一个有效的代理数字签名。因此，不论是原始签名者A还是代理签名者B，他们在生成一个数字签名后，不能再对它加以否认和抵赖。

⑤ 代理数字签名的可跟踪性。在本代理数字签名方案中，由于委托信息(SK_{ω},Q)是与原始签名者A及代理签名者B的身份绑定在一起的，因此，如果原始签名者A在向代理签名者B发送委托信息(SK_{ω},Q)的同时，将委托信息(SK_{ω},Q)与代理签名的身份一

起保存记录下来，那么原始签名者 A 事后可以根据某一有效的代理数字签名(s,Q)来确定是哪一个代理签名者签发了该消息，进而确定出相应的代理签名者的身份，实现对代理签名者的事后监督功能。通过对代理签名者所签发的代理数字签名进行跟踪，使得代理签名者不能在不被发现的情况下滥用它的代理签名权利，防止代理签名者滥用其所获得的代理签名权力。

⑥ 代理数字签名的密钥依赖性。在本代理数字签名方案中，由于代理签名私钥SK_o是原始签名者 A 的私钥SK_A的函数，显然，SK_o依赖于SK_A。

⑦ 代理数字签名密钥的可注销性。如果原始签名者A 想终止代 理签名者 B 的代理签名权力，注销代理签名者所拥有的代理数字 签名密钥SK_o，那么可以通过网络广播一条经过原始签名者 A 签名的系统消息，宣布Q不再有效，从而使得代理签名者B 所生成的所有代理数字签名全部随之失效，实现控制代理签名者的代理签名权力的目的。

总之，代理数字签名不仅真实地模拟了现实生活中的印章的全部功能，而且还具有可跟踪性、抗抵赖性等优秀特征，因而具有较为广泛的应用。

5.4.5 XECLPDS 受控代理数字签名方案

自代理签名体系出现以来，国内外学者在该领域进行了大量的研究，提出了许多代理签名方案。但所有这些方案都无法对代理签名者的代理数字签名权力进行完整的、可靠的控制。

本节要介绍一个由作者提出的、具有限制代理签名者行使代理签名权力的受控代理数字签名方案 XECLPDS(Xiao's Elliptic Curve Limited Proxy Digital Signature)。

在受控代理数字签名方案XECLPDS中，对椭圆曲线密码体系的基本参数有如下约定。

设椭圆曲线$E(GF(q))$是定义在有限域$GF(q)$上的一条安全椭圆曲线，其上随机选取的基点为G，设$n=\#E(GF(q))$是椭圆曲线E的阶，r是n的大数因子。设通信主体的私钥SK为小于$r-1$的随机正整数，公钥$PK=SK\times G$，并被置于可信的第三方认证中心CA处。

受控代理数字签名方案XECLPDS的工作过程如图5-12所示。由该图可以看出，受控代理数字签名方案XECLPDS同样由委托过程、签名过程和验证过程等三个核心过程组成。

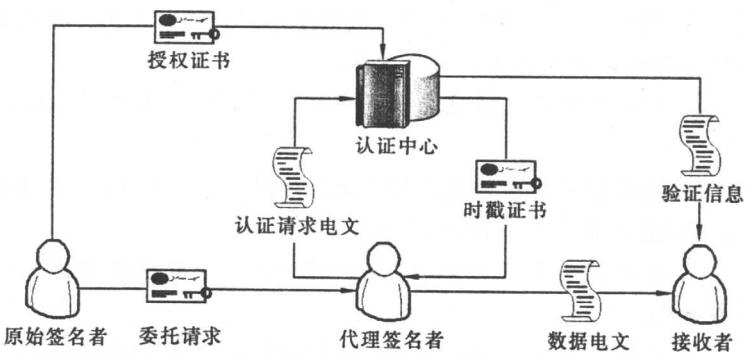

图5-12　受控代理数字签名的工作过程

为了完成对消息m的代理数字签名，原始签名者A和代理签名者B执行如下协议。

1. 委托过程

当原始签名者A因某种原因，需要授权代理签名者B在指定范围内代表原始签名者A行使A的数字签名权力时，执行下列操作。

① 原始签名者生成用于限制代理签名者的签名权限的授权文书A_p，其中包括约定的认证中心CA、有关代理签名者身份说明的电子证书、委托授权的有效期限、代理签名者代表原始签名者行使数字签名权力的权限范围、代理签名者代表原始签名者行使数字签名权力的最大签名次数等内容；

② 原始签名者A利用自己的私钥SK_A对限制代理签名者的签名权限的授权文书A_p进行普通数字签名，得到用于限制代理签名者的签名权限的授权证书C_p；

③ 原始签名者A将授权证书C_p发送给认证中心CA；

④ 认证中心CA在认证了所收到的授权证书合法性的基础上，根据授权证书的内容，在内部数据库中设定代理人代表委托人行使代理签名权力的最大签名次数；

⑤ 原始签名者A随机选取正整数$k\in[1,r]$，计算委托参数$Q_p=k\times G$；

⑥ 原始签名者A根据某种约定的杂凑摘要算法，计算信息串(C_p,Q_p)的杂凑摘要值$H_p=Hash(C_p,Q_p)$；

⑦ 原始签名者A根据自己的私钥SK_A，计算授权参数$S_p=H_p\times SK_A+k$；

⑧ 原始签名者A根据委托参数和授权参数，生成委托授权信息$M_p=(S_p,Q_p)$；

⑨ 原始签名者A将授权证书C_p和委托授权信息M_p绑定在一起，作为委托请求(C_p,M_p)，发送给代理签名者B。

代理签名者B在收到由原始签名者A所发送的委托请求后，执行下列操作，验证委托请求的合法性，决定是否接受委托。

① 代理签名者B从委托请求中析出授权证书C_p，利用原始签名者A的公钥PK_A，验证授权证书C_p的合法性；若授权证书C_p是有效的，则进行下一步；否则，要求原始签名者A重新发送委托请求，

或者直接拒绝该委托请求。

② 代理签名者B从授权证书C_p中析出授权文书A_p，并根据授权文书A_p的内容，决定是否接受原始签名者A的委托请求。如果接受原始签名者A的委托请求，则进行下一步。

③ 代理签名者B从委托请求中析出委托授权信息M_p，并从委托授权信息M_p中析出委托参数Q_p和授权参数S_p。

④ 代理签名者B根据某种约定的杂凑摘要算法，计算信息串(C_p, Q_p)的杂凑摘要值$H_p=H(C_p, Q_p)$。

⑤ 代理签名者B验证等式$S_p \times G = Q_p + H_p \times PK_A$是否成立。若等式不成立，则说明该委托授权信息无效，应要求原始签名者A重新发送委托请求；若等式成立，则说明该委托授权信息是有效的，予以接受，并可根据授权范围代表原始签名者A行使数字签名权力。

⑥ 代理签名者B根据从所收到的委托授权信息M_p中析出的授权参数S_p，以及计算所得到的杂凑摘要值H_p，计算代理签名私钥$SK_p=S_p+SK_B \times H_p$。

至此，原始签名者A完成了授权代理签名者B在指定范围内代表原始签名者A行使数字签名权力的委托过程。

2. 签名过程

当代理签名者B需要在自己的代理权限内代表原始签名者A对消息m行使数字签名权力时，需要与认证中心CA进行交互，执行下列操作。

① 代理签名者B利用代理签名私钥SK_p和所收到的委托授权信息M_p，按照某种约定的普通数字签名方法，对需要签署的消息m进行普通数字签名，生成普通数字签名$S'=Sig(SK_p, m, M_p)$，则代理签名者B代表原始签名者A对消息m生成的代理数字签名为

$S=(S', Q_p)$。

② 代理签名者B生成请求认证文书，其中包括：代理签名者B的身份信息、上一步中所得到的代理数字签名S以及请求认证中心CA对所签署的代理数字签名进行公证的认证消息。

③ 代理签名者B利用自己的私钥SK_B签署请求认证文书，得到请求认证电文，并将该电文发送给认证中心CA，请求认证。

④ 认证中心CA在收到代理签名者B提交的认证请求电文后，首先确认该电文的完整性和真实性。若有错误，则拒绝提供认证服务，并要求代理签名者B重新发送。若认证通过，则进行下一步操作。

⑤ 认证中心CA在内部数据库中搜索由原始签名者A设定的最大签名次数，以及代理签名者B已经代表原始签名者A行使数字签名权力的签名次数，如果在委托授权的最大签名次数的范围内，则根据收到认证请求电文的时间，对所收到的认证请求电文中的代理数字签名签发时戳证书T_p，同时修改内部数据库中的签名次数信息；否则，将拒绝提供认证服务，并终止操作。

⑥ 认证中心CA将所签署的时戳证书T_p返还给代理签名者B。

⑦ 代理签名者B在收到由认证中心签署的时戳证书T_p之后，将该证书附在消息之后，作为其代表原始签名者A对消息m所签署的代理数字签名的时间证明，与消息m、代理数字签名S、授权证书C_p绑定在一起，形成完整的数据电文$M=(m, S, C_p, T_p)$。

3. 验证过程

当电文接收者需要检验所收到的由代理签名者B代表原始签名者A签署的数据电文M的合法性时，执行如下验证步骤。

① 电文接收者从所收到的数据电文M中析出消息m、代理数

字签名S、授权证书C_p和时戳证书T_p。

② 电文接收者利用原始签名者A的公钥PK_A，验证授权证书C_p的合法性。利用认证中心CA的公钥PK_{CA}，验证时戳证书T_p的合法性。若验证不通过，则说明该数据电文无效。

③ 电文接收者根据授权证书C_p以及时戳证书T_p，检查代理签名者B是否有权签署该电子文书，即是否在限定的签名范围、限定的授权时效、限制的签名次数内进行的代理数字签名行为。若检查不通过，则说明该数据电文无效。

④ 电文接收者从代理数字签名S中析出委托参数Q_p，并根据某种约定的杂凑摘要算法，计算信息串(C_p, Q_p)的杂凑摘要值$H_p = Hash(C_p, Q_p)$。

⑤ 电文接收者根据代理数字签名S以及委托人的公钥PK_A和代理人的公钥PK_B，计算代理数字签名公钥$PK_p = Q_p + H_p \times (PK_A + PK_B)$。

⑥ 电文接收者从代理数字签名S中析出普通数字签名参数S'，利用事先约定的普通数字签名的签名验证方程验证所收到的代理数字签名的合法性，即判断方程

$$Ver(S', PK_p, m) = \text{True}$$

是否成立。若方程成立，则该代理数字签名是合法的；反之，则该代理数字签名是不合法的。

该代理数字签名方案的正确性证明如下：

$$\begin{aligned} Q_p + H_p \times (PK_A + PK_B) &= k \times G + H_p \times (SK_A + SK_B) \times G \\ &= (k + H_p \times SK_A) \times G + H_p \times SK_B \times G \\ &= (S_p + SK_B \times H_p) \times G \\ &= SK_p \times G \\ &= PK_p \end{aligned}$$

4. 代理数字签名的基本性质

(1) 基本的不可伪造性

由于代理签名者B无法根据所收到的委托请求(C_p, M_p)计算出原始签名者A的签名私钥SK_A,因此不可能伪造原始签名者A的普通数字签名。进而可知其他攻击者都不能生成原始签名者A的普通数字签名。

(2) 代理数字签名的不可伪造性

由于只有代理签名者B才能根据自己的签名私钥SK_B计算出代理签名私钥SK_p,进而生成消息m的代理数字签名,所以,除了代理签名者B外,任何人(包括原始签名者A)都不能伪造代理签名者B的有效代理数字签名。特别地,即使原始签名者A委托了多个代理签名者$B_1, B_2, \cdots, B_n$,那么任一代理签名者都不能伪造其他代理签名者的有效代理数字签名。

(3) 代理数字签名的可区分性

代理签名者B代表原始签名者A对消息m生成的代理数字签名(S', Q_p)是由普通数字签名S'和椭圆曲线上的某一数点Q_p两部分组成的。由于代理数字签名比普通数字签名多出一部分,显然,将代理数字签名和普通数字签名区分开很容易。由于数点Q_p是按公式$Q_p = k \times G$随机生成的,因此,不同代理签名者所生成的代理数字签名之间也可以被明显区分。

(4) 代理数字签名的不可抵赖性

由于任何人都不能伪造原始签名者A的普通数字签名,所以,A不能否认它的一个有效的普通数字签名。同理,由于除了代理签名者B外,任何人(包括原始签名者A)都不能伪造代理签名者B的有效代理数字签名,B也不能否认一个有效的代理数字签名。因

此，不论是原始签名者A还是代理签名者B，他们在生成一个数字签名后，不能再对它加以否认和抵赖。

(5) 代理数字签名的可跟踪性

在本代理数字签名方案中，由于委托请求(C_p, M_p)是与原始签名者A及代理签名者B的身份绑定在一起的，因此，原始签名者A可以根据某一有效的代理数字签名(S', Q_p)来确定是哪一个代理签名者签发了该消息，确定出相应的代理签名者的身份，实现对代理签名者的事后监督功能。

(6) 代理数字签名的密钥依赖性

在本代理数字签名方案中，由于代理签名私钥SK_p是原始签名者A的私钥SK_A的函数，显然，SK_p依赖于SK_A。

(7) 代理数字签名密钥的可注销性

在本代理数字签名方案中，原始签名者A可以通过授权证书C_p实现对代理签名者B代表原始签名者A行使数字签名的权力进行全面的、完整的和可靠的控制。

除此之处，本方案还具有下面的优点：代理的不可转移性和强可控制性。本方案中，即使代理签名者B将所得到的来自原始签名者A的授权委托请求直接转移给第三者C，那么由于原始签名者A的授权证书C_p中包含了代理签名者B的身份信息，因此，C不可能代表原始签名者A行使数字签名的权力。

另一方面，如果代理签名者B在所得到的授权之外滥用代理签名权力，那么，由于授权证书C_p和第三方可信认证中心CA的存在，代理签名者B所生成的任何代理数字签名都可以被确认是非法的。

综上所述，本小节所阐述的代理数字签名方案不仅符合我国《电子签名法》所规定对电子数字签名的完整性、抗抵赖性、抗伪造性等基本要求，而且具备很强的不可伪造性、很好的识别性和很强

的不可否认性等优秀性质以及防止代理人滥用代理权力的能力。与之前的各种类型代理数字签名方案相比，本方案能够实现对数字签名权力进行全面完整、安全可靠的委托授权。原始签名者能够严格地控制代理签名者行使数字签名权力的范围、时效、最大签名次数等，满足各种针对代理数字签名权力委托和授权问题的复杂需求，大大扩展了代理数字签名的应用范围，具有很好的实用价值。

5.4.6 其他数字签名方案

随着Internet 的迅猛发展，数字签名业已成为基于Internet 的各种活动（如电子商务、电子政务、电子出版等）必不可少的安全基础。为了适应各特定领域对数字签名的特殊需求，新的数字签名方案不断提出，除了上面所研究的几种基本的签名方案以外，还有下面一些特殊的签名方案。

1. 门限数字签名

在某些情况下，需要由一组用户来共同进行数字签名，以说明该消息为该组用户中的多数用户所共同认可，这时的数字签名就是门限数字签名（Threshold Digital Signature）。门限数字签名是一种基于“秘密共享”思想的数字签名方案，它的生成必须由多个成员合作才能完成，但它的验证只需要知道群体的公开密钥即可进行。在门限数字签名方案中，最著名的是Desmedt 等人提出的(t,n)门限数字签名方案。

在(t,n)门限数字签名方案中，n 个成员各自拥有整个群体的部分签名密钥，这使得任何多于t 个成员的子集可以代表群体产生签名，而任何少于t 个成员的子集则不能产生签名。也就是说，

为了给消息M产生一个有效的签名，至少需要$t+1$个群组成员合作。

门限数字签名方案具有两个重要的特征：门限特性和健壮性。在该方案中，设置了一个门限t，对于超过门限t的部分签名，其正确性是可以得到验证的。因此，该方案可以应用于电子选举、电子投票等系统中。

对于门限数字签名方案而言，其所面临的威胁主要来自群组内部，即由组内多个成员共同发起的“合谋攻击”。

2. 群数字签名

与门限数字签名方案类似，群数字签名（Group Digital Signature）方案也是一种涉及一组用户的数字签名方案。群数字签名方案允许一个群组中的成员以其所在的群组的名义签发一条消息，它同时具有下列三个特征。

① 只有群组的成员才能代表其所在的那个群组签发消息；

② 签名的接收者虽然可以验证该签名确实是来自那个群组的一个合法签名，却不能揭示该签名究竟是群组中的哪一个成员签发的；

③ 在必要的时候，借助于可信第三方机构或者多数群组成员，能够识别出群组中的签名者。

群数字签名方案主要由用于产生群数字签名的签名算法、识别签名合法性的验证算法以及识别签名者的识别算法等组成。

在群数字签名方案中，每个群组成员均拥有自己的签名密钥，由这些签名密钥中的任何一个所签发的消息，均可以用群组的公钥通过验证算法得到验证。在需要的时候，可以通过识别算法识别出群组中真正的签名者。

群数字签名方案具有有限匿名性，因而特别适用于电子投标

应用。在电子投标应用中，所有参加投标的公司组成一个群组，每个公司都可以使用自己的签名密钥匿名地签发他的投标。当特定的投标被选中后，由可信的第三方机构利用识别算法可以识别出中标者，而其他所有的投标者仍是匿名的；另一方面，若某一投标者有反悔、欺诈等行为，其身份能够得到识别。

除了门限数字签名方案和群数字签名方案以外，其他涉及一组用户的基本数字签名方案还有群定向数字签名方案和多重数字签名方案。群定向数字签名方案实际上是门限数字签名方案和群数字签名方案的结合，它允许群组的某个子集代表整个群组对消息进行签名，但不能识别出具体的签名者；多重数字签名则是指由多个用户对同一条消息进行联合签名。

3. 收方不可否认数字签名（Undeniable Digital Signature）

在普通的数字签名方案中，发送者是不能否认自己曾经发送过的消息的，而对接收方却没有任何约束，这样，就可能存在两种情况。

① 接收方已经阅读了消息，事后却否认自己曾接收过该消息。如接收方接收并阅读到了一条对自己不利的消息，然后，将它销毁，并否认自己曾经接收过该消息。

② 接收方故意拖延阅读时间，以做出对自己更有利的决定。如有甲、乙两公司，约定在某日上午10点之前由甲将合同文本发给乙，乙签字后发回给甲，然后合同生效。现在甲在10点之前将合同发给了乙，而乙此时想反悔，故意在10点之后才开始看合同，结果事后乙反而指责甲违约。

收方不可否认数字签名方案就是为了解决这些问题而提出来的。在这一方案中，没有签名者的合作，接收方就无法验证签名，这

使得发送者的利益得到某种程度上的保护。

一个收方不可否认数字签名方案是由签名协议、验证协议和否认协议三种协议组成的。其中，否认协议(Disavoael Protocol)是核心部分，它通过让发送者和接收方执行否认协议完成一致性检测来推断签名的真伪。若有一方拒绝参加否认协议，则可以判定欺诈行为的存在。

当涉及与时间有关的通信时，还需要引入可信第三方和时戳服务器作为仲裁者，记录双方通信的时间和阅读情况，以确定责任归属。这时通信的双方虽然需要由可信第三方对双方的通信进行担保，但却不希望其获知通信的具体内容，因此还需要使用盲签名技术。

4. 面向版权保护的数字签名技术

由于数字签名具有身份鉴别、数据认证和抗否认的功能，因而，又被用于数字版权保护，作为版权拥有者的版权声明和验证。

对数字媒体的版权进行签名主要有两种方式。一种是对某一数字文件(如图像、软件等)进行签名，由于这类文件的大小通常是确定的，因而其签名方案与普通的数字签名方案一样。另一种是对流媒体(如在线广播、在线音乐、在线视频、在线电影等)进行签名，这类文件的数据长度很长，甚至于无限长，而接收方是无法事先确定消息流长度的，因此不可能像普通的数字签名方案中要求的那样，在收到全部消息之后再对签名进行验证。这样，面向流信息的流数字签名方案就应运而生。

流数字签名方案一般采用基于有向图的签名算法，只需要对一段数据流中的某一个关键数据包进行实时签名和验证，就可以保证整段数据流的安全。这种技术目前已经应用于Real System公司的Real Server和Microsoft公司的Media Server之上。

除了用于版权保护以外，面向流信息的签名技术还能应用于其他领域，比如，当通信代价较高时，接收方可先对大型消息文件进行验证，然后再决定是否接收，以节约通信资源。

数字签名作为保障网络电子商务信息安全的重要技术之一，其应用日益广泛。数字签名具有身份鉴别、数据认证和抗否认等功能。与传统的书面签名相比，它不仅能够鉴别签名者的身份，而且还能证明消息的内容，保证了消息的完整性；此外，数字签名还能够显著地节省时间和资金、易于存储，降低了交易的风险。目前，数字签名已经在欧洲各国、美国和联合国内获得了与手写签名同等的法律地位，而中国首部有关数字签名的法规——《中国电子商务、电子政务法律环境及电子签章法》也在2003年底正式颁布实施，这些标志着数字签名技术将会拥有更加广阔的发展应用前景。

第6章　椭圆曲线密码体系的若干关键技术

自1985年基于椭圆曲线离散对数问题的椭圆曲线公钥密码体系被发明以来，至今为止，已经出现了多种椭圆曲线公钥密码体系。从上一章可以知道，在所有这些密码编码体系的实现中，有两种类型的基本运算：一类是在密码体系设计阶段所要用到的基本运算，另一类是密码体系运行阶段所要用到的基本运算。

第一类基本运算涉及椭圆曲线密码体系基本参数的选取，它包括安全椭圆曲线的寻找和基点的选取两个部分。

分析上一章所介绍的各种椭圆曲线公钥密码体系可知，第二类基本运算主要是椭圆曲线有限群上的各种代数运算，它包括点加运算、倍点运算和数乘运算等三种运算。与第一类基本运算相比，第二类基本运算决定了椭圆曲线公钥密码体系的运行效率。因而人们对第二类基本运算的实现算法的工作效率要求更高，需要精心设计。

6.1　寻找安全椭圆曲线

本书的第3.5节讨论了安全椭圆曲线的选取准则，归纳了各种安全曲线的类型。从长远角度看，为了应付各种可能的针对椭圆曲线密码体系的攻击，在各种安全椭圆曲线中，笔者倾向于选择通

过随机方式生成的具有大素数阶的理想安全椭圆曲线作为椭圆曲线密码体系的基础。

在第 3.5 节，介绍了得到一条随机性很好的安全椭圆曲线的一般方法，其过程大致如下：首先需要根据基域的种类和相应的椭圆曲线方程，随机选取曲线的参数；然后利用 SEA 算法计算椭圆曲线有限群的阶 $\#E(GF(q))$；当求出阶 $\#E(GF(q))$ 之后，再对所求得的 $\#E(GF(q))$ 进行分解，进行素性测试，直到 $\#E(GF(q))$ 为素数或其中含有一个理想的大素数因子，否则需要继续重新随机选取新的曲线参数。这样才能得到比较理想的随机性好的安全椭圆曲线。这一过程如算法 6.1 所示。

算法 6.1　生成随机性好的安全椭圆曲线

输入：基域特征 $p>3$
输出：安全椭圆曲线方程 $y^2=x^3+ax+b, a,b\in F$

1. 随机生成椭圆曲线参数 $a,b\in GF(p)$；
2. 检测参数 a,b 是否合法：$ab\neq 0$，$\Delta=4a^3+27b^2\neq 0$；若不然，返回第 1 步，重新选取新的曲线参数 a,b；
3. 用 SEA 算法计算 $\#E(GF(p))$；
4. 对 $\#E(GF(p))$ 进行素性测试，若不理想，则返回第 1 步，重新选取新的曲线参数 a,b；
5. 输出曲线方程 $y^2=x^3+ax+b$。

显然，这一过程比较复杂，曲线的生成效率比较低、速度比较慢。分析算法 6.1 后可以发现，事实上，并不需要在完全计算出曲线的阶 $\#E(GF(q))$ 之后才能通过素性测试判断其是否合适。

由第 4.7 节知，在 SEA 算法的第②阶段，需要收集有关 $t \bmod l$ 的一系列信息，若能在这一阶段判断出 l 是 $\#E(GF(q))$ 的一个因

子，则可以直接得出结论，判定$\#E(GF(q))$为合数，从而提前进入下一轮选取和测试过程。

对于Elkies素数，由于能够求出$t \bmod l$的精确值，这样，若l是$\#E(GF(q))$的一个因子，则必有

$$\#E(GF(q)) = q + 1 - t \equiv 0 \bmod l$$

所以，l是$\#E(GF(q))$的一个因子，与$q+1 \equiv t \bmod l$等价。其具体算法如下。

算法6.2　生成理想的安全椭圆曲线

输入：基域特征$p>3$

输出：安全椭圆曲线方程$y^2=x^3+ax+b, a,b \in F$

1. 随机生成椭圆曲线参数$a,b \in GF(p)$；
2. 检测参数a,b是否合法：$ab \neq 0, \Delta = 4a^3+27b^2 \neq 0$；若不然，返回第1步，重新选取新的曲线参数$a,b$；
3. 对Atkin素数，由Atkin方法计算T_l；
4. 对Elkies素数，当$l<I_m$时，用同种圈方法计算$t \bmod l^n$的精确值；
   4.1　若$p+1 \equiv t \bmod l^n$，则$\#E(GF(p))$是合数，返回第1步，重新选取新的曲线参数a,b；
5. 其他的Elkies素数，用Elkies方法计算$t \bmod l$的精确值；
   5.1　若$p+1 \equiv t \bmod l$，则$\#E(GF(p))$是合数，返回第1步，重新选取新的曲线参数a,b；
6. 对$\#E(GF(p))$进行素性测试，若为合数，则返回第1步，继续选取新的曲线参数a,b；
7. 输出曲线方程$y^2=x^3+ax+b$。

比较算法6.1和算法6.2后可知，算法6.2的效率比算法6.1

高得多，在相同的时间内，算法 6.2 可以检测更多的椭圆曲线。故此，选用算法 6.2 作为随机安全椭圆曲线的生成算法，用 Borland 公司的 Delphi 6.0 实现了一个高随机性安全椭圆曲线生成系统，该系统的运行结果如下。

① 随机选取的素数有限域 $GF(p)$，其特征值为

$p=74946395712905314625782128651408252769239211375 1$

② 随机选取的安全椭圆曲线参数

$a=45362670968510060554381883210008902913624823037 1$

$b=21640965503242462340557763143835439959563098847 7$

③ 用 SEA 算法求出的该椭圆曲线的阶为

$r=37473197856452657312891016330636993103122062473 3$

同时可知，它是一个 159 位的素数。

④ 系统同时根据第 6.2 节所介绍的算法 6.3 选取了基点 $G=(x,y)$，

$x=73330541903188229099262328743677781938653672057 3$

$y=69234298943940391644454558679529967552435941843 6$

6.2 基点的选取

从数学的观点看，基点是椭圆曲线有限加法群的大素因子子群的一个随机选择的生成元。从椭圆曲线密码体系看，基点是现有各种椭圆曲线密码体系中的重要参数，是用户密钥对产生的基础。由于基点在椭圆曲线密码体系中的重要性，一旦选定，一般应该保持其稳定性，不能随便更换。基点虽然是随机产生的，但为了保证系统的安全性，一般要求所选择的基点必须具有大素数阶。

据此，可以给出如下所示的素域上基点选取算法。

算法6.3　基点选取算法

输入：椭圆曲线E，有限域$GF(p)$，且$p>3$
输出：E上的一个随机的非零基点$G=(x,y)$

1. 随机选择$0\leqslant x<p$；
2. $\alpha\leftarrow x^3+ax+b \bmod p$；
3. 若$\alpha=0$，则输出$G=(x,0)$，退出；
4. 应用逐点降幂法求解平方剩余问题$\beta^2=\alpha \bmod p$；
   4.1　若$\beta^2=\alpha \bmod p$无解，返回第1步，重新选择x；
   4.2　若β有解，则产生随机位μ，置$y\leftarrow(-1)^{\mu}\beta \bmod p$；
5. 检查G点的阶，若不符合要求，则重新选择；
6. 输出$G=(x,y)$。

在求解有限域$GF(p)$上的平方剩余问题$\beta^2=\alpha \bmod p$时，可以根据p值的情况，利用逐点降幂法实现快速求解。

由有限域中关于平方剩余的Euler判据可知，若整数α为奇素数p的某一个平方剩余，则必有$\alpha^{(p-1)/2}\equiv 1 \bmod p$。若整数$\alpha$是奇素数$p$的非平方剩余，则有$\alpha^{(p-1)/2}\equiv -1 \bmod p$。下面分两种情况讨论。

① 当$p\equiv 3 \bmod 4$时，对$\alpha=x^3+ax+b$，若$\beta^2=\alpha \bmod p$有解，则有$\alpha^{(p-1)/2}\equiv 1 \bmod p$，在方程两边同时乘以$\alpha$，则有

$$\alpha^{(p+1)/2}\equiv \alpha \bmod p \tag{6.1}$$

因为$p\equiv 3 \bmod 4$，$\dfrac{p+1}{4}$是整数，则

$$\alpha^{(p+1)/2}=(\alpha^{(p+1)/4})^2 \tag{6.2}$$

由式(6.1)和式(6.2)易知

$$(\alpha^{(p+1)/4})^2\equiv \alpha \bmod p$$

所以有

$$\beta = \alpha^{(p+1)/4} \bmod p \tag{6.3}$$

② 当 $p \equiv 1 \bmod 4$ 时，类似地，若 $\beta^2 = \alpha \bmod p$ 有解，则有 $\alpha^{(p-1)/2} \equiv 1 \bmod p$，所以

$$\alpha^{(p-1)/2} - 1 \equiv 0 \bmod p \tag{6.4}$$

由于 $p \equiv 1 \bmod 4$，$\frac{p-1}{4}$ 是整数，对方程式(6.4)分解可得

$$(\alpha^{(p-1)/4} - 1)(\alpha^{(p-1)/4} + 1) \equiv 0 \bmod p \tag{6.5}$$

显然，只有当 $\alpha^{(p-1)/4} \equiv \pm 1 \bmod p$ 时，方程 $\beta^2 = \alpha \bmod p$ 有解，否则无解。

若 $\alpha^{(p-1)/4} \equiv 1 \bmod p$，则在方程 $\beta^2 = \alpha \bmod p$ 两边同时乘以 α，有

$$(\alpha^{(p+3)/8})^2 = \alpha \times \alpha^{(p-1)/4} \equiv \alpha \bmod p$$

所以

$$\beta = \alpha^{(p+3)/8} \bmod p \tag{6.6}$$

若 $\alpha^{(p-1)/4} \equiv -1 \bmod p$，则由有限域中的互倒定理和平方剩余的性质有

$$2^{(p-1)/2} \equiv -1 \bmod p \tag{6.7}$$

所以

$$\alpha^{(p-1)/4} \equiv 2^{(p-1)/2} \bmod p \tag{6.8}$$

等式两边同时乘以 $2^{(p-1)/2}\alpha$，有

$$2^{(p-1)/2}\alpha^{(p+3)/4} \equiv 2^{p-1}\alpha \bmod p$$

由 $2^{p-1} \equiv 1 \bmod p$，有

$$(2^{(p-1)/4}\alpha^{(p+3)/8})^2 \equiv \alpha \bmod p \tag{6.9}$$

所以

$$\beta = 2^{(p-1)/4} \times \alpha^{(p+3)/8} \bmod p \tag{6.10}$$

因此，根据奇素数 p 和 $\alpha^{(p-1)/4}$ 的情况，从方程式(6.3)、方程式

(6.6)和方程式(6.10)中选择合适的公式即可求出β或者判定平方剩余方程$\beta^2=\alpha \bmod p$无解。

将随机基点选取算法并入算法6.2,可以实现一个完整的理想椭圆曲线密码体系的参数生成系统。

6.3　基本群运算的实现

椭圆曲线密码体系的核心是椭圆曲线离散对数问题。绝大多数椭圆曲线密码体系都是将密钥或明文消息通过某种变换嵌入到椭圆曲线群中的某一群元(点)来实现的。所以,在椭圆曲线密码体系的设计和实现中,需要考虑椭圆曲线群上基本群运算的快速实现。

对椭圆曲线密码体系而言,数乘运算是它的核心运算,所以,提高其运算速度是本节的主要目的。另一方面,数乘运算又依赖于点加运算、倍点运算两种基本群运算。

由2.5节可知,对任意一条椭圆曲线E而言,可以有两种坐标系来表示其上的点:其一是仿射坐标系,这时椭圆曲线E的Weierstrass方程为方程式(2.6);其二是射影坐标系,这时椭圆曲线E的Weierstrass方程为方程式(2.7)。用不同的坐标表示,会影响这两种基本群运算的运行速度。本节将分别介绍在这两种坐标表示下的两种群运算的快速实现算法。

椭圆曲线E上的群运算是按照椭圆曲线群运算法则,在定义椭圆曲线的有限域$GF(q)$内完成的,其中所涉及的运算包括加、减、乘、求逆和平方五种基本运算。这五种运算中,加法与减法相对其他运算来说,所用时间可以忽略不计。

为方便对各种算法的运行性能的分析，用M表示在有限域$GF(q)$中的一次乘法运算，用S表示在有限域$GF(q)$中的一次平方运算，用I表示在有限域$GF(q)$中的一次求逆运算。这里，将平方运算与乘法运算分开的原因在于平方运算可以使用快速求解平方剩余问题的逐点降幂算法，在普通微机上的实验表明

$1S \approx 0.8M$。

分析点加运算和倍点运算需要分两种坐标系讨论。

1. 仿射坐标系下的点加运算与倍点运算

由3.1节，这里根据有限域$GF(q)$的特征值情况，分别处理。

① 特征$p>3$，且为素数时。由方程式(2.14)，Weierstrass方程形式为

$$y^2 = x^3 + ax + b, \quad a,b \in F$$

且系数a,b满足条件

$$\Delta(E) = 4a^3 + 27b^2 \bmod p \neq 0$$

设椭圆曲线E上任意两个非零点$P=(x_1,y_1)$，$Q=(x_2,y_2)$，且有$P \neq -Q$，设$R=(x_3,y_3)=P+Q$，则由椭圆曲线群的群运算法则有

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases}$$

其中

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{当 } P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1}, & \text{当 } P = Q \end{cases}$$

算法6.4为仿射坐标系下的点加运算和倍点运算算法。

算法6.4　仿射坐标系下的点加和倍点运算

输入：素数$p>3$，椭圆曲线E，以及$P=(x_1,y_1)$，$Q=(x_2,y_2)$
输出：点$R=(x_3,y_3)$，$R=P+Q$

1. 若$P=O$，则输出$R\leftarrow P$，并停止；
2. 若$Q=O$，则输出$R\leftarrow Q$，并停止；
3. 若$x_1\neq x_2$，则
   3.1 $\lambda\leftarrow(y_2-y_1)/(x_2-x_1) \bmod p$；
   3.2 转至第7步；
4. 若$y_1\neq y_2$，则输出$R\leftarrow O$，并停止；
5. 若$y_2=0$，则输出$R\leftarrow O$，并停止；
6. $\lambda\leftarrow 3(x_1^2+a)/(2y_1) \bmod p$；
7. $x_3\leftarrow\lambda^2-x_1-x_2 \bmod p$；
8. $y_3\leftarrow(x_1-x_3)\lambda-y_1 \bmod p$；
9. 输出$R\leftarrow(x_3,y_3)$。

由算法6.4可知，对于点加运算，需要在基域上完成1次求逆运算、1次平方运算和2次乘法运算，其运算量为$1I+1S+2M$。类似地，对倍点运算，需要在基域上完成1次求逆运算、2次平方运算和2次乘法运算，其运算量为$1I+2S+2M$。

② 特征$p=2$时，有限域为$GF(2^n)$。根据安全椭圆曲线的选取准则可知，椭圆曲线E是非奇异型的，所以，由方程式(2.15)，Weierstrass方程形式为

$$y^2+xy=x^3+ax^2+b,a,b\in F$$

且系数a,b满足条件

$$\Delta(E)=b\neq 0$$

设椭圆曲线E上任意两个非零点$P=(x_1,y_1)$，$Q=(x_2,y_2)$，且

有$P\neq -Q$，设$R=(x_3,y_3)=P+Q$，则由椭圆曲线群的群运算法则有

$$x_3=\begin{cases}\lambda^2+\lambda+x_1+x_2+a, & \text{当 } P\neq Q\\ \lambda^2+\lambda+a, & \text{当 } P=Q\end{cases}$$

$$y_3=(x_2+x_3)\lambda+x_3+y_2$$

其中，

$$\lambda=\begin{cases}\dfrac{y_1+y_2}{x_1+x_2}, & \text{当 } P\neq Q\\ x_1+\dfrac{y_1}{x_1}, & \text{当 } P=Q\end{cases}$$

故对二元有限域上椭圆曲线的点加运算和倍点运算有算法6.5。

算法6.5　仿射坐标系下的点加和倍点运算

输入：域$GF(2^n)$，椭圆曲线E，以及$P=(x_1,y_1)$，$Q=(x_2,y_2)$

输出：点$R=(x_3,y_3)$，$R=P+Q$

1. 若$P=O$，则输出$R\leftarrow P$，并停止；
2. 若$Q=O$，则输出$R\leftarrow Q$，并停止；
3. 若$x_1\neq x_2$，则
   3.1　$\lambda\leftarrow(y_1+y_2)/(x_1+x_2)$，$x_3\leftarrow\lambda^2+\lambda+x_1+x_2+a$；
   3.2　转至第7步；
4. 若$y_1\neq y_2$，则输出$R\leftarrow O$，并停止；
5. 若$x_2=0$，则输出$R\leftarrow O$，并停止；
6. 置$\lambda\leftarrow x_1+y_1/x_1$，$x_3\leftarrow\lambda^2+\lambda+a$；
7. $y_3\leftarrow(x_2+x_3)\lambda+x_3+y_2$；
8. 输出$R\leftarrow(x_3,y_3)$。

由算法6.5可知，对于二元有限域上的仿射坐标下的椭圆曲线群上的点加运算和倍点运算，均需要在基域上完成1次求逆运算、1次平方运算和2次乘法运算，其运算量均为$1I+1S+2M$。

2. 射影坐标系下的点加运算与倍点运算

设椭圆曲线 E 上任意两个非零点 $P=(X_1,Y_1,Z_1)$，$Q=(X_2,Y_2,Z_2)$，则 $P\neq -Q$，设 $R=(X_3,Y_3,Z_3)=P+Q$。显然，有许多种变换公式能够完成仿射坐标与射影坐标之间的变换，这里所要介绍的Jacobian 带权射影坐标变换表示则是其中最快的。具体变换公式如下。

$$x_i=\frac{X_i}{Z_i^2},\quad y_i=\frac{Y_i}{Z_i^3}\quad (i=1,2,3) \tag{6.11}$$

显然，对任意的非零 $\lambda\in GF(q)$，有

$$(X,Y,Z)=(\lambda^2X,\lambda^3Y,\lambda Z)$$

而无穷远点 O 为 $(\lambda^2,\lambda^3,0)$，其中 $\lambda\neq 0$。

要将射影坐标下的点 (X_i,Y_i,Z_i) 变换为仿射坐标下的点 (x_i,y_i) 很简单，只需令 $Z_i=1$ 即可。

利用变换公式(6.11)，可以给出射影坐标下的椭圆曲线群运算法则。与仿射坐标下的情况类似，这里也根据有限域的情况，分两种情况讨论。

① 特征 $p>3$，且为素数时，由方程式(2.14)和变换式(6.11)，可知 Weierstrass 方程形式为

$$Y^2=X^3+aXZ^4+bZ^6 \tag{6.12}$$

其系数 a,b 满足条件

$$\Delta(E)=4a^3+27b^2 \bmod p\neq 0$$

由仿射坐标下的群运算法则方程式(3.2)、方程式(3.3)和变换式(6.11)，可得射影坐标下的椭圆曲线群运算法则如下。

● 当 $P\neq Q$ 时，将式(6.11)分别代入式(3.2)、式(3.3)，可得

$$\begin{cases}\dfrac{X_3}{Z_3^2}=\left(\dfrac{Y_2Z_1^3-Y_1Z_2^3}{X_2Z_1^3Z_2-X_1Z_1Z_2^3}\right)^2-\dfrac{X_1}{Z_1^2}-\dfrac{X_2}{Z_2^2}\\ \dfrac{Y_3}{Z_3^3}=\left(\dfrac{Y_2Z_1^3-Y_1Z_2^3}{X_2Z_1^3Z_2-X_1Z_1Z_2^3}\right)\left(\dfrac{X_1}{Z_1^2}-\dfrac{X_3}{Z_3^2}\right)-\dfrac{Y_1}{Z_1^3}\end{cases}$$

令$U_1=X_1Z_2^2$,$S_1=Y_1Z_2^3$,$U_2=X_2Z_1^2$,$S_2=Y_2Z_1^3$,整理上式可得

$$\begin{cases}X_3=R^2-TW^2\\ Y_3=(VR-MW^3)/2\\ Z_3=Z_1Z_2W\end{cases}\tag{6.13}$$

其中 $W=U_1-U_2$, $R=S_1-S_2$, $T=U_1+U_2$

$M=S_1+S_2$, $V=TW^2-2X_2$

由此对射影坐标系下的点加运算有算法6.6。

算法6.6 射影坐标系下的点加运算

输入:素数$p>3$,椭圆曲线E,以及$P=(X_1,Y_1,Z_1)$,$Q=(X_2,Y_2,Z_2)$

输出:点$R=(X_3,Y_3,Z_3)$,$R=P+Q$

1. $T_1\leftarrow X_1$[若$Z_2=1$,则该值为U_1];
2. $T_2\leftarrow Y_1$[若$Z_2=1$,则该值为S_1];
3. $T_3\leftarrow Z_1$;
4. $T_4\leftarrow X_2$;
5. $T_5\leftarrow Y_2$;
6. 若$Z_2\neq 1$,则
   - 6.1 $T_6\leftarrow Z_2$, $T_7\leftarrow T_6^2$;
   - 6.2 $T_1\leftarrow T_1\times T_7$[当$Z_2\neq 1$,计算$U_1$];
   - 6.3 $T_7\leftarrow T_6\times T_7$;
   - 6.4 $T_2\leftarrow T_2\times T_7$[当$Z_2\neq 1$,计算$S_1$];

7. $T_7 \leftarrow T_3^2$;
8. $T_4 \leftarrow T_4 \times T_7$[计算$U_2$];
9. $T_7 \leftarrow T_3 \times T_7$;
10. $T_5 \leftarrow T_5 \times T_7$[计算$S_2$];
11. $T_4 \leftarrow T_1 - T_4$[计算W];
12. $T_5 \leftarrow T_2 - T_5$[计算R];
13. 若$T_4=0$,则
    13.1　若$T_5=0$,则$P=Q$,返回(0,0,0),并停止;
    13.2　否则,返回$R=O=(1,1,0)$,并停止;
14. $T_1 \leftarrow 2 \times T_1 - T_4$[计算$T$];
15. $T_2 \leftarrow 2 \times T_2 - T_5$[计算$M$];
16. 若$Z_2 \neq 1$,则$T_3 \leftarrow T_3 \times T_6$;
17. $T_3 \leftarrow T_3 \times T_4$[计算$Z_3$];
18. $T_7 \leftarrow T_4^2$, $T_4 \leftarrow T_4 \times T_7$, $T_7 \leftarrow T_1 \times T_7$;
19. $T_1 \leftarrow T_5^2$, $T_1 \leftarrow T_1 - T_7$[计算X_3];
20. $T_7 \leftarrow T_7 - 2 \times T_1$[计算$V$];
21. $T_5 \leftarrow T_5 \times T_1$, $T_4 \leftarrow T_2 \times T_4$, $T_2 \leftarrow T_5 - T_4$, $T_2 \leftarrow T_2/2$[计算Y_3];
22. $X_3 \leftarrow T_1$, $Y_3 \leftarrow T_2$, $Z_3 \leftarrow T_3$,输出$R=(X_3,Y_3,Z_3)$。

由算法6.6可知,对于点加运算,需要在基域上完成4次平方运算和12次乘法运算,其运算量为$4S+12M$。特别地,在数乘运算中,有$Z_1=1$,$Z_2=1$,则只需在基域上完成2次平方运算和8次乘法运算,故其运算量为$2S+8M$。

● 当$P=Q$时,将式(6.11)分别代入方程式(3.2)、方程式(3.3),可得

$$\begin{cases}\dfrac{X_3}{Z_3^2}=\left(\dfrac{3X_1^2+aZ_1^4}{2Y_1Z_1}\right)^2-2x_1\\ \dfrac{Y_3}{Z_3^3}=\left(\dfrac{3X_1^2+aZ_1^4}{2Y_1Z_1}\right)\left(\dfrac{X_1}{Z_1^2}-\dfrac{X_3}{Z_3^2}\right)-\dfrac{Y_1}{Z_1^3}\end{cases}$$

整理可得

$$\begin{cases}X_3=M^2-2S\\ Y_3=M(S-X_3)-T\\ Z_3=2Y_1Z_1\end{cases}\tag{6.14}$$

其中，$M=3X_1^2+aZ_1^4$，$S=4X_1Y_1^2$，$T=8Y_1^4$。

若$a\equiv(p-3) \bmod p$，由于

$$\begin{aligned}M&=3X_1^2-3Z_1^4\\&=3(X_1^2-Z_1^4)\\&=3(X_1+Z_1^2)(X_1-Z_1^2)\end{aligned}$$

则可减少两次基域上的平方运算。

由此，对射影坐标系下的倍点运算有算法6.7。

算法6.7　射影坐标系下的倍点运算

输入：素数$p>3$，椭圆曲线E，以及$P=(X_1,Y_1,Z_1)$

输出：点$R=(X_3,Y_3,Z_3)$，$R=2P$

1. $T_1\leftarrow X_1$，$T_2\leftarrow Y_1$，$T_3\leftarrow Z_1$；
2. 若$T_2=0$或$T_3=0$，则返回$R=O=(1,1,0)$，并停止；
3. 若$a=p-3$，则
   3.1　$T_4\leftarrow T_3^2$；
   3.2　$T_5\leftarrow T_1-T_4$，$T_4\leftarrow T_1+T_4$；
   3.3　$T_5\leftarrow T_4\times T_5$，$T_4\leftarrow 3\times T_5$；
   否则

3.1　$T_4 \leftarrow a$；

3.2　$T_5 \leftarrow T_3^2, T_5 \leftarrow T_5^2, T_5 \leftarrow T_4 \times T_5$；

3.3　$T_4 \leftarrow T_1^2, T_4 \leftarrow 3 \times T_4, T_4 \leftarrow T_4 + T_5$；

4. $T_3 \leftarrow T_2 \times T_3, T_3 \leftarrow 2 \times T_3$[计算$Z_3$]；

5. $T_2 \leftarrow T_2^2$；

6. $T_5 \leftarrow T_1 \times T_2, T_5 \leftarrow 4 \times T_5$[计算$S$]；

7. $T_1 \leftarrow T_4^2$；

8. $T_1 \leftarrow T_1 - 2 \times T_5$[计算$X_3$]；

9. $T_2 \leftarrow T_2^2$；

10. $T_2 \leftarrow 8 \times T_2$[计算$T$]；

11. $T_5 \leftarrow T_5 - T_1$；

12. $T_5 \leftarrow T_4 \times T_5$；

13. $T_2 \leftarrow T_5 - T_2$[计算Y_3]；

14. $X_3 \leftarrow T_1, Y_3 \leftarrow T_2, Z_3 \leftarrow T_3$，输出$R=(X_3, Y_3, Z_3)$。

由算法6.7可知，对于倍点运算，需要在基域上完成6次平方运算和4次乘法运算，其运算量为$6S+4M$。特别地，当a足够小的时候，可以用累加运算来减少一次乘法运算，此时运算总量为$6S+3M$；当$a \equiv (p-3) \bmod p$时，则可减少两次平方运算，故运算总量为$4S+4M$。

Chudnovsky指出，由于椭圆曲线之间存在着同构关系，对素数有限域$GF(p)$而言，当$p \equiv 3 \bmod 4$时，有1/2的椭圆曲线可以转化为$a \equiv (p-3) \bmod p$的情况；而当$p \equiv 1 \bmod 4$时，也有1/4的椭圆曲线可以转化为$a \equiv (p-3) \bmod p$的情况，所以，$a \equiv (p-3) \bmod p$的快速倍点算法是很有意义的。

② 特征$p=2$时，有限域为$GF(2^m)$，安全椭圆曲线E是非奇异

型的，此时，由方程式(2.15)和变换式(6.11)，得 Weierstrass 方程形式为

$$Y^2 + XYZ = X^3 + aX^2Z^2 + bZ^6 \tag{6.15}$$

且系数 a,b 满足 $\Delta(E)=b\neq 0$。

类似地，分两种情况讨论射影坐标下的椭圆曲线群运算法则。

● 当 $P\neq Q$ 时，将方程式(6.11)分别代入方程式(3.5)、方程式(3.6)和方程式(3.7)，可得

$$\begin{cases}\dfrac{X_3}{Z_3^2}=\left(\dfrac{Y_1Z_2^3+Y_2Z_1^3}{X_1Z_1Z_2^3+X_2Z_1^3Z_2}\right)^2+\left(\dfrac{Y_1Z_2^3+Y_2Z_1^3}{X_1Z_1Z_2^3+X_2Z_1^3Z_2}\right)+\dfrac{X_1}{Z_1^2}+\dfrac{X_2}{Z_2^2}+a\\ \dfrac{Y_3}{Z_3^3}=\left(\dfrac{X_2}{Z_2^2}+\dfrac{X_3}{Z_3^2}\right)\left(\dfrac{Y_1Z_2^3+Y_2Z_1^3}{X_1Z_1Z_2^3+X_2Z_1^3Z_2}\right)+\dfrac{X_3}{Z_3^2}+\dfrac{Y_2}{Z_2^3}\end{cases}$$

令 $U_1=X_1Z_2^2, S_1=Y_1Z_2^3, U_2=X_2Z_1^2, S_2=Y_2Z_1^3$，整理上式可得

$$\begin{cases}X_3 = aZ_3^2 + TR + W^3\\ Y_3 = TX_3 + VL^2\\ Z_3 = LZ_2\end{cases} \tag{6.16}$$

其中 $W=U_1+U_2,\quad R=S_1+S_2,\quad L=Z_1W$

$$V=RX_2+LY_2,\quad T=R+Z_3$$

算法 6.8 能完成射影坐标表示下的二元有限域上椭圆曲线群的点加运算。

算法 6.8　射影坐标系下的点加运算(二元有限域)

输入：域 $GF(2^n)$，椭圆曲线 E，以及 $P=(X_1,Y_1,Z_1)$， $Q=(X_2,Y_2,Z_2)$ 输出：点 $R=(X_3,Y_3,Z_3)$，$R=P+Q$
1. $T_1\leftarrow X_1$[若 $Z_2=1$，则该值为 U_1]； 2. $T_2\leftarrow Y_1$[若 $Z_2=1$，则该值为 S_1]； 3. $T_3\leftarrow Z_1$；

4. $T_4 \leftarrow X_2$;
5. $T_5 \leftarrow Y_2$;
6. 若$a \neq 0$,则$T_9 \leftarrow a$;
7. 若$Z_2 \neq 1$,则
   7.1　$T_6 \leftarrow Z_2, T_7 \leftarrow T_6^2$;
   7.2　$T_1 \leftarrow T_1 \times T_7$[当$Z_2 \neq 1$,计算$U_1$];
   7.3　$T_7 \leftarrow T_6 \times T_7$;
   7.4　$T_2 \leftarrow T_2 \times T_7$[当$Z_2 \neq 1$,计算$S_1$];
8. $T_7 \leftarrow T_3^2$;
9. $T_8 \leftarrow T_4 \times T_7$[计算$U_2$];
10. $T_1 \leftarrow T_1 + T_8$[计算W];
11. $T_7 \leftarrow T_3 \times T_7$;
12. $T_8 \leftarrow T_5 \times T_7$[计算$S_2$];
13. $T_2 \leftarrow T_2 + T_8$[计算R];
14. 若$T_1 = 0$,则
    14.1　若$T_2 = 0$,则$P = Q$,返回(0,0,0),并停止;
    14.2　否则,返回$R = O = (1,1,0)$,并停止;
15. $T_4 \leftarrow T_2 \times T_4$;
16. $T_3 \leftarrow T_1 \times T_3$[计算$L$;若$Z_2 = 1$,则该值为$Z_3$];
17. $T_5 \leftarrow T_3 \times T_5$;
18. $T_4 \leftarrow T_4 + T_5$[计算V];
19. $T_5 \leftarrow T_3^2$;
20. $T_7 \leftarrow T_4 \times T_5$;
21. 若$Z_2 \neq 1$,则$T_3 \leftarrow T_3 \times T_6$[若$Z_2 \neq 1$,计算$Z_3$];
22. $T_4 \leftarrow T_2 + T_3$[计算T];
23. $T_2 \leftarrow T_2 \times T_4$;

24. $T_5 \leftarrow T_1^2$;
25. $T_1 \leftarrow T_1 \times T_5$;
26. 若 $a \neq 0$,则
    26.1 $T_8 \leftarrow T_3^2$;
    26.2 $T_9 \leftarrow T_8 \times T_9$;
    26.3 $T_1 \leftarrow T_1 + T_9$;
27. $T_1 \leftarrow T_1 + T_2$[计算 X_3];
28. $T_4 \leftarrow T_1 \times T_4$;
29. $T_2 \leftarrow T_4 + T_7$[计算 Y_3];
30. $X_3 \leftarrow T_1, Y_3 \leftarrow T_2, Z_3 \leftarrow T_3$,输出 $R=(X_3, Y_3, Z_3)$。

由算法 6.8 可知,对于点加运算,需要在基域上完成 5 次平方运算和 15 次乘法运算,其运算量为 $5S+15M$。特别地,当 $a=0$ 时,只需要 4 次平方运算和 14 次乘法运算,运算总量为 $4S+14M$。由椭圆曲线的同构性可知,有一半的椭圆曲线可以转化为 $a=0$ 的情景,从而可以加速运算。

● 当 $P=Q$ 时,类似地,可得 Jacobian 射影坐标表示下的群运算公式为

$$\begin{cases} X_3 = (X_1 + cZ_1^2)^4 \\ Y_3 = X_1^4 Z_3 + UX_3 \\ Z_3 = X_1 Z_1^2 \end{cases} \tag{6.17}$$

其中

$$U = Z_3 + X_1^2 + Y_1 Z$$

域元素 $c=b^{2^{m-2}}$(即 $b=c^4$)。

算法 6.9 能描述了射影坐标表示下的二元有限域上椭圆曲线群的倍点运算。

由算法 6.9 可知,对于倍点运算,需要在基域上完成 5 次平方

运算和5次乘法运算，其运算总量为$5S+5M$。

算法6.9　射影坐标系下的倍点运算(二元有限域)

输入：域$GF(2^n)$，椭圆曲线E，以及$P=(X_1,Y_1,Z_1)$
输出：点$R=(X_3,Y_3,Z_3)$，$R=2P$

1. $T_1\leftarrow X_1$；
2. $T_2\leftarrow Y_1$；
3. $T_3\leftarrow Z_1$；
4. $T_4\leftarrow c$；
5. 若$T_1=0$或$T_3=0$，则返回$R=O=(1,1,0)$，并停止；
6. $T_2\leftarrow T_2\times T_3$；
7. $T_3\leftarrow T_3^2$；
8. $T_4\leftarrow T_3\times T_4$；
9. $T_3\leftarrow T_1\times T_3$[计算$Z_3$]；
10. $T_2\leftarrow T_2+T_3$；
11. $T_4\leftarrow T_1+T_4$；
12. $T_4\leftarrow T_4^2$；
13. $T_4\leftarrow T_4^2$[计算X_3]；
14. $T_1\leftarrow T_1^2$；
15. $T_2\leftarrow T_1+T_2$[计算U]；
16. $T_2\leftarrow T_2\times T_4$；
17. $T_1\leftarrow T_1^2$；
18. $T_1\leftarrow T_1\times T_3$；
19. $T_2\leftarrow T_1+T_2$[计算Y_3]；
20. $T_1\leftarrow T_4$；
21. $X_3\leftarrow T_1$，$Y_3\leftarrow T_2$，$Z_3\leftarrow T_3$，输出$R=(X_3,Y_3,Z_3)$。

上述两种坐标表示下的算法及其运算量对比结果如表6.1所示。

由表6.1可以看出，Jacobian射影坐标表示下的椭圆曲线群上的点加运算和倍点运算(算法6.6～算法6.9)不需要进行求逆运算。所以，如果在基域上完成求逆运算所需花费的时间远大于完成乘法运算所需要的时间，则用Jacobian射影坐标表示可以大大加快椭圆曲线群上的点加运算和倍点运算。也就是说，在具体的实现算法中，采用何种坐标系来完成上述点加运算和倍点运算，依赖于基域中求逆运算和乘法运算的速度比。

表6.1 两种坐标表示下的算法及运算量比较

有限域类型	群运算种类	仿射坐标		射影坐标	
		算法	运算量	算法	运算量
$GF(p)$	点加	算法6.4	$1I+1S+2M$	算法6.6	$2S+8M$
	倍点	算法6.4	$1I+2S+2M$	算法6.7	$4S+4M$
$GF(2^m)$	点加	算法6.5	$1I+1S+2M$	算法6.8	$4S+14M$
	倍点	算法6.5	$1I+1S+2M$	算法6.9	$5S+5M$

具体来说，在素数有限域$GF(p)$中，由于求逆运算相当慢，其运行时间超过乘法运算时间的10倍以上，因而一般采用Jacobian射影坐标来表示椭圆曲线，以提高系统的运行性能。

考虑到二次有限域的运行速率低于素域，所以，不是在特殊的情况下，一般都选择素数有限域作为椭圆曲线密码系统的基域。

1998年，Cohen提出了一种混合坐标表示，它能够综合上述两种坐标表示的优点，进一步提高了运行速度。

6.4 椭圆曲线有限群上的数乘运算

由2.7节可知，椭圆曲线群中的数乘运算mP是普通Abel有限乘法群中的模指数幂运算g^a的特例，现有的一切针对模幂运算的快速算法和技巧都可以应用于椭圆曲线群的数乘运算中。本节将首先给出几种对任何Abel群幂指运算都有效的快速算法，然后在此基础上分析椭圆曲线群中求逆运算几乎不需要时间的特点，给出上述这些算法的改进版本，最后得出椭圆曲线有限群上数乘运算的快速算法。

1. 一般Abel群上的幂指运算的实现算法

设有限Abel群G中的群运算用加法表示，则由定义2.9可知，群G上的幂指运算g^a可以用ag来表示。现设$P\in G$，$\#G\approx O(2^n)$，且整数$m\in[1,\#G]$，则有很多算法能用于快速求解群G上的数乘运算mP，例如，

① 二进制算法，也称“平方-乘”算法；

② k进制算法及其改进版本——滑动窗口算法。

所有这些算法都是通过采用不同的方式重新表示m来实现减少计算量的。在这些算法中，最基本的当属“平方-乘”二进制算法，为此，首先介绍它，然后再介绍其他算法。

(1) 二进制算法

现设整数m的二进制表示为

$$m=\sum_{i=0}^{l}a_i2^i \tag{6.18}$$

其中，$i=0,1,\cdots,l$，$a_i=0$或1，且最高位$a_l=1$。

对于数乘运算mP，有

$$Q = mP = \left(\sum_{i=0}^{l} a_i 2^i\right) P$$

$$= 2(\cdots 2(2(2P + a_{l-1}P) + a_{l-2}P) + \cdots)P + a_0 P$$

令$Q_0 = P$，可得迭代公式为

$$\begin{cases} Q_1 = 2Q_0 + a_{l-1}P \\ Q_2 = 2Q_1 + a_{l-2}P \\ \vdots \\ Q_l = 2Q_{l-1} + a_0 P \end{cases} \tag{6.19}$$

显然，$Q = Q_l$。利用二进制算法求解mP时，共需完成l次倍点运算和$W(m)-1$次点加运算。其中，$W(m)$为用二进制方法表示m时的Hamming重量（即表达式(6.18)的系数集合$\{a_i\}$中非零元素总数）。

(2) *k*进制算法及其改进

设r是一正整数，取$k = 2^r$，并设整数m的k进制表示为

$$m = \sum_{i=0}^{l} a_i k^i \tag{6.20}$$

其中，$i = 0, 1, \cdots, l$；$a_i = 0, \cdots, k-1$。

则对于数乘运算mP，有

$$Q = mP = \left(\sum_{i=0}^{l} a_i k^i\right) P$$

$$= k(\cdots k(k(ka_l P + a_{l-1}P) + a_{l-2}P) + \cdots)P + a_0 P$$

令$Q_0 = a_l P$，可得迭代公式为

$$\begin{cases} Q_1 = kQ_0 + a_{l-1}P \\ Q_2 = kQ_1 + a_{l-2}P \\ \vdots \\ Q_l = kQ_{l-1} + a_0 P \end{cases} \tag{6.21}$$

显然，$Q=Q_l$。

由此可知，用k进制算法求解mP时，需要分两个阶段。

第一阶段：预计算阶段。依次计算$2P,3P,\cdots,(k-1)P$，并列表存储，为下一阶段利用迭代式(6.21)直接计算a_iP做准备，这一阶段需要完成$(k-2)$次点加运算。

第二阶段：正式计算阶段。设群元$Q\in G$，$k=2^r$，这里称数乘运算kQ为k倍点运算，显然，一个数乘运算kQ相当于r次倍点运算。类似地，用$W(m)$表示用k进制方法表示m时的Hamming重量，有$W(m)<l$。

因此，利用第一阶段的预计算结果和迭代式(6.21)，能够在l次k倍点运算和$W(m)$次点加运算后求出mP。

考虑预计算阶段的工作量，在利用k进制算法计算mP时，共需完成rl次倍点运算和$W(m)+(k-2)$次点加运算。与二进制算法相比，在计算mP时所需要的倍点运算次数相当，但点加运算次数减少了。

由迭代式(6.21)可知，在利用k进制算法计算mP时，基本运算式为

$$kQ + a_iP = 2^rQ + a_iP \tag{6.22}$$

为简化计算，设

$$a_i = 2^{s_i}h_i, i = 0,1,\cdots,l-1$$

其中，$s_i<r$，奇数$h_i<m$。代入式(6.22)，得

$$\begin{aligned} kQ + a_iP &= 2^rQ + 2^{s_i}h_iP \\ &= 2^{s_i}(2^{r-s_i}Q + h_iP) \end{aligned}$$

将式(6.22)中的系数a_i中2的因子分离出来与2^rQ合并计算。这样，由于h_i是奇数，所以可以减少大约一半的预计算阶段的计算量。

为此，可将预计算阶段过程改进如下。

首先，计算倍点 $2P$ 的值，然后，利用递归公式

$$(2i+1)P = (2i-1)P + 2P$$

依次计算出 $3P, 5P, \cdots, (k-1)P$，并列表存储，这时全部的预计算量为 1 次倍点运算和 $\frac{k-2}{2}$ 次点加运算。

正式计算阶段的迭代公式为

$$\begin{cases} Q_1 = 2^{s_{l-1}}(2^{r-s_{l-1}}Q_0 + h_{l-1}P) \\ Q_2 = 2^{s_{l-2}}(2^{r-s_{l-2}}Q_1 + h_{l-2}P) \\ \quad \vdots \\ Q_l = 2^{s_0}(2^{r-s_0}Q_{l-1} + h_0P) \end{cases} \tag{6.23}$$

显然，与式(6.21)对比可知，式(6.23)的计算量没有变化。故改进后的 k 进制算法在计算 mP 时，总共减少了 $\frac{k-2}{2}$ 次点加运算。

对比二进制平方-乘算法和 k 进制算法可知，k 进制算法实际上是在二进制算法的基础上，通过每隔 r 项合并计算来达到减少计算量的目的。但是，由于式(6.18)中系数 a_i 取 0 或取 1 是完全随机的，因此，按固定间隔 r 来合并计算的效果有时并不好。为解决这一问题，Blake，Menezes 等人提出了不固定合并间隔 r 的大小，根据 a_i 取 0 或取 1 的具体情况来灵活处理的改进 k 进制算法，又称滑动窗口算法。

2. 椭圆曲线有限群上的数乘运算的快速实现算法

对于定义在有限域 $GF(q)$ 上的椭圆曲线有限群 E，设点 $P \in E(GF(q))$，对任一整数 $m \in [1, \#E(GF(q))]$，在采用上面的几种算法求解 mP 时，所需完成的倍点运算次数基本不变，但所需要的点加运算次数与 m 的某一类似于式(6.20)的表示式中的系数集合 $\{a_i\}$ 中的非零元素的个数有关。$\{a_i\}$ 中的非零元素的个数越少，所

需要完成的点加运算次数也越少。而在椭圆曲线有限群中，由公式(2.19)可知，求逆元运算几乎不花费时间，因而减法运算和加法运算的耗时是相同的。这样，可以考虑对m的某一表示式进行重新编码，允许系数a_i取-1，以期增加$\{a_i\}$中零元素的个数，进而减少所需的点加运算次数。

为此，先引入下面一些定义。

定义6.1 对任意整数m，设$m=\sum_{i=0}^{l-1}a_i2^i, a_i\in\{-1,0,1\}$，则称$(a_{l-1}a_{l-2}\cdots a_1a_0)_2$为整数$m$的有符号二进制表示，也称扩展二进制表示，并记$-1=\bar{1}$。

显然，任何整数都可以表示为有符号二进制串，且其表示不唯一。利用有符号二进制串对m进行重新编码的目的就是使得m的某一有符号二进制表示中非零元素个数尽可能减少。

定义6.2 对整数m的某一有符号二进制表示$(a_{l-1}a_{l-2}\cdots a_1a_0)_2$，若没有相邻的非零元素，即任何相邻的两个系数中至少有一个为0，亦即对于所有的i，有$a_{i+1}\times a_i=0$，称该表示为全不连通型(Non-Adjacent Form，NAF)的有符号二进制表示。

关于m的NAF表示，Morain和Olivor进行了证明。

定理6.1 对每一个整数m，存在唯一的NAF表示，其长度比m的二进制表示最多多1，且在m的所有的有符号二进制表示中，该NAF表示的Hamming重量最小，亦即其中的非零元素个数最少；对长度为l的NAF，其Hamming重量的期望值为$l/3$。

于是，对椭圆曲线群E中的数乘运算mP，若系数m采用NAF编码，则可大大减少运算过程中所需的点加运算次数。而由表6.1可知，在椭圆曲线有限群中，点加运算所花费的时间并不少于同等条件下的倍点运算所花费的时间。这样，利用NAF编码可以大大加快数乘运算的实现速度。

为了得到整数m的NAF表示，有算法6.10。

算法6.10 NAF编码算法

输入：整数 $m=\sum_{i=0}^{l-1} m_i 2^i, m_i \in \{0,1\}$

输出：NAF编码 $m=\sum_{i=0}^{l} a_i 2^i, a_i \in \{-1,0,1\}$

1. $c_0 \leftarrow 0$;
2. $m_l \leftarrow 0$;
3. $m_{l+1} \leftarrow 0$;
4. For j from 0 to l do;
   4.1 $c_{j+1} \leftarrow [(m_j+m_{j+1}+c_j)/2]$; 4.2 $a_j \leftarrow m_j+c_j-2c_{j+1}$;
5. 输出$(a_l a_{l-1}\cdots a_1 a_0)_2$。

观察NAF的编码算法，发现有一种更简单的描述方式：对某一正整数k，令$h=3k$，设$h=(h_l h_{l-1}\cdots h_1 h_0)_2$，$k=(k_l k_{l-1}\cdots k_1 k_0)_2$分别为$h$和$k$的无符号二进制表示，并在集合$\{(0,0),(0,1),(1,0),(1,1)\}$与集合$\{(-1,0,1)\}$，即$\{(\bar{1},0,1)\}$之间建立如下映射关系

$$\begin{cases}(0,0)\rightarrow 0\\(1,1)\rightarrow 0\\(0,1)\rightarrow \bar{1}\\(1,0)\rightarrow 1\end{cases} \tag{6.24}$$

则二元组序列$(h_l,k_l)(h_{l-1},k_{l-1})\cdots(h_1,k_1)$可按照映射关系式(6.24)映射为$k$的全不连通型NAF的有符号二进制表示。显然，映射关系式(6.24)可用h_i-k_i来表示。

于是，对椭圆曲线群E中的数乘运算mP，首先利用算法6.10对m进行NAF编码，然后即可套用前面介绍的二进制平方-乘算

法、k 进制算法、滑动窗口算法等方法快速计算 mP。

下面，以二进制平方-乘算法为例，给出针对椭圆曲线有限群上数乘运算的NAF 快速算法。

算法 6.11 NAF 数乘算法

输入：整数 m，$P \in E$ 为椭圆曲线群上的点
输出：$Q=mP$

1. 若 $m=0$，输出 O 并返回 Q；
2. 若 $m < 0$，则置 $P \leftarrow (-P)$，$n \leftarrow (-n)$；
3. 求 k 的二进制表示 $k=(k_l k_{l-1} \cdots k_1 k_0)_2$；
4. 求 $3k$ 的二进制表示 $h=(h_l h_{l-1} \cdots h_1 h_0)_2$；
5. $Q \leftarrow P$；
6. For i from $l-1$ downto 1 do [计算迭代式(6.18)]
   6.1 $Q \leftarrow 2Q$；
   6.2 若 $k_i=0$，且 $h_i=1$，计算 $Q \leftarrow Q+P$；[$a_i=1$]
   6.3 若 $k_i=1$，且 $h_i=0$，计算 $Q \leftarrow Q-P$；[$a_i=-1$]
7. 输出 Q。

利用NAF 二进制数乘算法求解 mP 时，需要花费 l 次倍点运算和平均 $l/3$ 次点加运算。与普通二进制平方-乘算法相比，NAF 二进制算法具有更高的效率。而有关NAF 与其他的如 k 进制算法、滑动窗口算法等算法的具体结合方法，这里暂不介绍。

第7章　椭圆曲线密码体系的实践

第5章和第6章从应用角度介绍和分析了各种椭圆曲线密码体系方案以及实现中的若干关键问题。本章将在此研究基础上，继续深入研究椭圆曲线公钥密码体系具体实现的问题，并给出椭圆曲线公钥密码体系中各种密码方案的具体实现算法和实验结果，包括任意长度安全真随机密钥生成方法XRNGS、XKAS密钥协商与管理方案、XKDS密钥分配方案、X-ElGamal加密算法和XHES混合密码加密算法、XECDS-I普通签名方案和XECBDS盲数字签名方案等。

7.1　任意长度安全真随机密钥的生成

在5.2.1节中，研究了椭圆曲线公钥密码体系中用户基本密钥对的生成问题，其中关键之一是用户私有密钥的生成。本节将重点介绍任意长度安全真随机密钥的生成方法，着重介绍作者提出的真随机密钥生成方法XRNGS（Xiao's Random Number Generate Scheme）。

在信息安全系统中，密钥是合法访问的唯一凭证。Kerckhoff假设指出：一个信息安全系统的安全性取决于该系统所采用的密钥本身，而不在于系统自身或者通信硬件的安全保护程度。也就是

说，除了密钥信息以外，其他的一切信息都可以是公开的。在这一前提下，对于一个信息安全系统而言，它所选用的密码体系和算法本身可以被公开，访问策略可以公布，密码设备可能丢失，但该信息安全系统仍然可以继续使用，不受影响。但是，密钥一旦泄漏，整个安全系统将被破坏，不但合法用户不能访问系统、提取信息，而且系统中的信息将会被非法用户所窃取，进而危害到整个系统的安全。由此可见，安全的密钥管理方案在通信系统的安全中是十分关键和极其重要的。它不仅影响着系统的安全性，而且还将涉及系统的可靠性、有效性和经济性等内容。

由Shanlon定理可知，安全的密钥要求具有尽可能高的熵值。而要取得较高的熵值，则要求所选择的密钥具有一定的长度和高度的随机性，并且能避免可预测性。

1. 常规密钥产生法

现代通信网络的高速发展，人们对大批量的安全密钥的需求越来越高。目前，已经出现了许多各种各样的密钥产生器，用于产生各种密钥。按其密钥产生的工作原理，可以分成以下三类。

(1) 某一随机算法控制法

这一类密钥产生方法是目前最常见的一类密钥产生方法，它依据某一事先确定的随机算法或者随机数表产生密钥。但是，由于算法自身的特征(对于给定的输入，有确定的输出)，从理论上讲，所产生的密钥是可以被预测的。

例如，1999年发现的针对电子支付协议SSL的攻击就是从SSL临时会话密钥的产生算法中预测出由其产生的所有可能密钥的组合，从中寻找和分析出真正的密钥，进而突破电子商务支付系统。因此，这类密钥产生器被称为伪随机密钥产生器，它只能用在对安全性要求不高的场合。

(2) 人工法

这类方法通过掷硬币、扔骰子等随机方式获得密钥，是目前被公认的最安全的密钥产生方法之一。由这类方法产生的密钥具有较高的熵值，有高度的随机性，并能够避免可预测性。

但是这类方法使用起来非常繁琐。例如，为了利用掷硬币来产生一个商业安全标准所需要的 1 024 位长度的密钥，就需要掷 1 024次硬币，很不实用，不能适应现代社会对密钥产生量的需求，因此，除了在极少数非常重要的场合，一般不用这种方法来获得安全密钥。

(3) 随机噪声发生检测法

这类密钥产生方法是通过产生、测量自然界中具有高度随机性的噪声信号，以获得所需要的具有较高的熵值、均匀的外部特征、完全没有规律可循、能够避免可预测性的高质量的量子力学意义上真随机密钥。因而需要极其昂贵的随机噪声振荡器、随机信号发生器等硬件设备。目前这类真随机信号发生装置大多是通过产生和检测放射性衰变、微弱放射线、粒子轨迹、半导体热噪声、石英振荡器等噪声源来获得较为理想的真随机密钥。由于这些装置的结构复杂、操作繁琐、价格昂贵、产生随机密钥的速度较慢，而且具有一定的危险性，因此，这类方法既不方便，也不很实用。

综上所述，现有的三类密钥产生方法都存在着各种各样的问题，有的安全性不够、有的不方便、还有的不实用。所有这些问题都导致高随机性的安全密钥的产生效率极低，密钥的管理难度加大、密钥的更换频率降低等，为信息系统的安全带来了巨大的隐患。

2. 真随机密钥生成法

在分析现有的各种密钥产生方法的基础上，作者提出了真随机密钥生成方法 XRNGS(Xiao's Random Number Generate

Scheme)。该方法通过实时跟踪测量鼠标器、触摸屏、触摸板等定点设备的移动情况，利用定点设备随机移动时方位、速度和指点频率等各种参量的随机变化情况，获得安全可靠的真随机密钥。该方法的原理如图7-1所示。

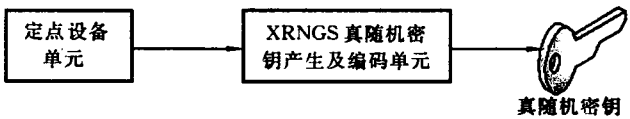

图7-1　XRNGS真随机密钥产生方法原理框图

众所周知，最简单、最常见的定点设备就是鼠标。所以，这里给出一种最简单的基于鼠标器的具体实施方式，其步骤如下。

(1) 用户随机移动鼠标

通过跟踪鼠标器的移动情况，对其进行离散采样，以获取该鼠标在随机移动时所产生的方位、速度和指点频率等各种参量的离散随机变化情况。具体方法如下。

① 方位参量的获取。设上一次采样时，鼠标器的指点坐标为(x_1, y_1)；而本次采样时，鼠标器的指点坐标为(x_2, y_2)。可以依据两次采样所得到的指点坐标之间的关系和方程式(7.1)，确定鼠标器当前的移动方位。

$$Direction = \frac{y_2 - y_1}{x_2 - x_1} \tag{7.1}$$

由此，可得到方位参量的获取算法7.1。

算法7.1　方位参量的获取算法

输入：两次采样所得到的鼠标器指点坐标(x_1, y_1)，(x_2, y_2)
输出：鼠标器的移动方位 *Direction*

1. 若 $x_1=x_2$，输出 O 并退出；
2. 求 $u=y_2-y_1, v=x_2-x_1$；
3. 计算 $Direction=\frac{u}{v}$；
4. 返回 $Direction$。

显然，该算法的时间复杂度为 $O(1)$。

② 速度参量的获取。设上一次采样时刻为 t_1，鼠标器的指点坐标为 (x_1, y_1)；而本次采样时刻为 t_2，鼠标器的指点坐标为 (x_2, y_2)。可以依据两次采样所得到的指点坐标和两次采样的时间间隔之间的关系和式(7.2)，确定鼠标器当前的移动速度。

$$Velocity=\sqrt{\left(\frac{x_2-x_1}{t_2-t_1}\right)^2+\left(\frac{y_2-y_1}{t_2-t_1}\right)^2} \tag{7.2}$$

由此，可得到速度参量的获取算法 7.2。

算法 7.2　速度参量的获取算法

输入：两次采样所得到的鼠标器指点坐标 (x_1, y_1)，(x_2, y_2)，
　　两次采样的时刻 t_1 和 t_2
输出：鼠标器的移动速度 $Velocity$

1. 计算 $\Delta t=t_2-t_1$；[两次采样的时间间隔]
2. 求 $u=y_2-y_1, v=x_2-x_1$；
3. 计算 $V_x=\frac{v}{\Delta t}, V_y=\frac{u}{\Delta t}$；
4. 计算 $Velocity=\sqrt{V_x^2+V_y^2}$；
5. 返回 $Velocity$。

显然，该算法的时间复杂度为 $O(1)$。

③ 指点频率的获取。设两次采样时刻分别为t_1和t_2，鼠标器按键的指点计数分别为c_1和c_2。可以依据两次采样所得到的指点计数和两次采样的时间间隔之间的关系，依据式(7.3)，确定鼠标器当前的指点频率。

$$Frequency = \frac{c_2 - c_1}{t_2 - t_1} \tag{7.3}$$

由此，有指点频率的获取算法7.3。

算法7.3　指点频率的获取算法

输入：两次采样的时刻t_1和t_2，所得到的鼠标器指点计数c_1和c_2 输出：鼠标器的指点频率$Frequency$
1. 计算$\Delta t = t_2 - t_1$；[两次采样的时间间隔] 2. 求$Count = c_2 - c_1$； 3. 计算$Frequency = \frac{Count}{\Delta t}$； 4. 返回$Frequency$。

(2) *编码*

这一步是依据事先预定的随机密钥产生策略，对已获取的鼠标在随机移动时所产生的移动方位、移动速度和指点频率等各种参量的离散随机变化数据进行组合计算和编码，获得具有较高的熵值和高度的随机性以及难以预测的具有某一固定长度的高强度安全密钥。

重复上面的过程，直到所获得的密钥的总长度满足系统需求为止。

密码学意义上的随机密钥的质量检测指标包括频数检测、跟随特性检测、随机性检测、分布均匀性检测和独立性检测等多种指标。其中频数指标和跟随特性指标一般采用计数统计方法进行测

试，随机性指标一般使用 χ^2 方法进行测试，分布均匀性指标一般使用 χ^2 拟合优度法进行测试，独立性指标常用游程检验法进行测试。

在所有随机密钥质量检测方法中，以美国国家技术标准局(National Institute of Standards and Technology，NIST)于2001年5月发布的关于密码系统的信息安全标准FIPS 140-2(Federal Information Processing Standards Publication，FIPS)最为著名。在FIPS 140-2中指定了4种测试方式对随机密钥的质量指标进行测试，以取代常规的随机性统计检验，以合格区间的形式简化了随机密钥质量的检验过程。与其他类似的标准相比，FIPS 140-2标准更加严格。

FIPS 140-2标准中规定从随机序列中随机选取20 000位连续的位序列，进行下列四种测试。

① 单比特测试(the monobit test)。本项测试通过计算20 000位序列中1的个数(X)，当9 725$<X<$10 275时，则表示该项测试通过。

② 扑克测试(the poker test)。本项测试将20 000位的位序列按4位1组分成5 000组，每组(4位)有16种可能的取值，统计5 000组中每组可能的取值数字。设$f(i)$为取值为i的组数字，$i\in[0,15]$，则按下列公式(7.4)计算X。

$$X=\frac{16}{5\ 000}\times\sum_{i=0}^{15}[f(i)]^2-15 \tag{7.4}$$

若2.16$<X<$46.17，则表示通过该项测试。

③ 游程测试(the runs test)。一个游程定义为连续为0或1的最大位序列，其中0或1的个数称为游程的长度，统计20 000位的位序列中各种不同长度游程的数量。显然每种长度的游程有两种情况，若所有游程长度均满足表7.1所示标准，则通过该项测试。

表7.1 游程测试合格标准

游程长度	1	2	3	4	5	6
区间	2 343～2 657	1 135～1 365	542～708	251～373	111～201	111～201

④ 长游程测试(long runs test)。长游程定义为游程长度大于等于26的游程。若没有长游程,则表示测试通过。易知,该项测试等价于分别计算0和1的所有游程长度的最大值,若该值均小于26,则表示通过该项测试。

用Borland Delphi 6.0在Windows 2000平台上实现了XRNGS方法,并采用上述的FIPS 140-2标准对连续生成的500组真随机密钥进行了测试,测试结果全部合格,密钥输出速率为25 Kb/s。部分测试数据如表7.2所示。

表7.2 FIPS140-2测试数据

测试类型		游程长度	合法范围	测试结果I	测试结果II	测试结果III	测试结果IV	测试结果V
单比特测试			9 725～10 275	10 081	10 118	10 065	9 975	9 992
扑克测试			2.16～46.17	8.858	15.526	8.109	24.634	11.059
游程测试	0	1	2 343～2 657	2 409	2 539	2 512	2 409	2 453
	1	1	2 343～2 657	2 380	2 445	2 448	2 380	2 494
	0	2	1 135～1 365	1 258	1 210	1 214	1 258	1 267
	1	2	1 135～1 365	1 299	1 232	1 250	1 299	1 197
	0	3	542～708	642	636	643	642	594
	1	3	542～708	666	673	648	666	638
	0	4	251～373	319	322	298	319	321
	1	4	251～373	280	326	311	280	299
	0	5	111～201	143	129	154	143	159
	1	5	111～201	168	156	169	168	162
	0	≥6	111～201	175	156	158	175	168
	1	≥6	111～201	150	160	152	150	171
最大游程长度	0	<26	1～25	12	14	12	12	10
	1	<26	1～25	12	13	16	12	12

总之，XRNGS 方法结合了现有各种密钥产生方法的长处，避免了使用复杂的随机噪声信号发生振荡器，以及手工操作的繁琐过程，能够获得具有较高的熵值和高度的随机性以及难以预测的高强度安全密钥。它操作简单，经济实用，能够取代现有的各种密钥产生方法，适用于所有类型的定点设备及各种复杂的应用环境，每秒可以产生 25 Kb 的高质量的、满足信息安全标准的随机密钥，因而具有很好的实用价值。

在实际应用环境中，有时会碰到这样一类应用场合：应用环境中没有定点设备，同时又需要能快速地获得具有较高的熵值和高度的随机性，并难以预测的高强度安全密钥。

这时，研究的重点在于如何获取自然界中的噪声信号源。在比较简单的情况下，可以通过随机按键的情况和变化频率，或者通过加装位移传感器、速度传感器或压力传感器以及 A/D 转换器，来对各种随机变化参量进行离散采样，由类似于 XRNGS 方法中的密钥编码算法，即可获得具有较高的熵值和高度的随机性的高强度安全密钥。

如果应用环境比较复杂，那么就需要从其他途径获取自然界中的白噪声信号作为随机信号源。这里给出两种本人研究出的利用自然界中的白噪声信号源产生真随机密钥的方法及相关的装置。

方法一：利用自然界无线电电磁信号中的白噪声作为真随机密钥发生器的信号源，其原理框图如图 7-2 所示。

从图 7-2 可以看出，该方法利用自然界中的无线电电磁波信号中的白噪声作为真随机密钥产生装置的信号源，通过无线传感电路采集无线电电磁信号中的白噪声，利用 A/D 转换器对所采集的随机白噪声信号进行离散采样，由预设的真随机密钥编码算法即可获得具有较高的熵值和高度的随机性的高强度安全密

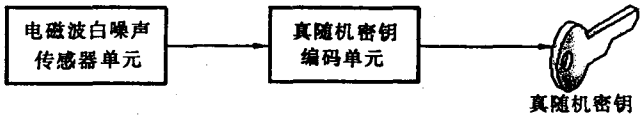

图 7-2　基于无线电电磁波白噪声信号的真随机密钥生成方法原理框图

钥。

由于电磁波白噪声传感器单元的功能在于采集广泛存在于自然界中的无线电电磁波信号中的白噪声，作为后继真随机密钥编码单元的随机信号源，因此，在实际应用中，可以选用简单的中波收音电路作为电磁波白噪声传感器单元。这里选择SONY 公司生产的CXA1238S 单芯收音电路作为电磁波白噪声传感器单元，并将9，10 号管脚短接，关闭自动增益和自动频率电路。

真随机密钥编码单元的功能在于根据从电磁波白噪声传感器单元所得到的随机信号源，按某种预设的真随机密钥编码算法获得具有较高的熵值和高度的随机性的高强度安全密钥。选择美国MAXIM 公司生产的16 位单电源、低功耗、单/双极性转换的高精度串行逐次逼近型模数转换器MAX1132。该芯片仅需单一+5 V 供电，同时带有内部基准参考电压和时钟，内置跟踪/保持及校准电路，而且内置串行输入/输出接口和编程接口，外围电路简单，因而可简化电路设计和控制设计，大大降低产品的成本。

与现有各种真随机密钥产生方法相比，本方法的优点在于所采用的装置本身不带噪声源，它利用自然界中无处不在的无线电电磁信号中的白噪声作为真随机密钥的噪声信号源，自身不需要附加复杂的随机噪声振荡发生源，完全不需要手工操作就能够快速地、自动地产生大量的、可靠安全的、具有较高熵值的、外部特征均匀的、高度的随机性、完全没有规律可循、难以预测的高质量的

真随机密钥，从而避免使用复杂、昂贵的诸如粒子轨迹检测、放射源衰变之类的真随机信号发生器，因此具有很好的实用价值。

方法二：利用自然界中的电力传输过程中所存在的电压、电流等电力线白噪声信号，以此作为真随机密钥发生器的信号源，其原理框图如图 7-3 所示。

图 7-3　基于电力线白噪声信号的真随机密钥生成方法原理框图

从图 7-3 可以看出，该方法利用自然界中的电力线白噪声信号作为真随机密钥产生装置的信号源，通过电力线互感传感电路采集电力线白噪声信号，利用 A/D 转换器对所采集的随机白噪声信号进行离散采样，由预设于嵌入式芯片中的真随机密钥编码算法即可获得具有较高的熵值和高度的随机性的安全密钥。

由于电力线白噪声传感器单元的功能在于采集随处可得的电力线白噪声信号作为后继真随机密钥编码单元的随机信号源，因此，在实际应用中，可以选用简单的耦合互感电路，通过对比电力线进行高通滤波前后的信号差来获取电力线白噪声。

真随机密钥编码单元的功能在于根据从电力线白噪声传感器单元所得到的随机信号源，按某种预设于嵌入式芯片中的真随机密钥编码算法获得具有较高的熵值和高度的随机性的高强度安全密钥。选择三星公司生产的嵌入式处理器S3C4510B 芯片，利用该芯片对所采集的电力线白噪声信号进行Hash 杂凑编码，即可获得稳定的真随机密钥。

7.2 XKAS密钥协商方案

密钥协商是密钥管理中的重要课题，它能够让通信系统中的两个或多个参与主体在一个公开的、不安全的信道上通过通信协商联合建立一个只在本次通话或数据交换过程中有效的临时会话密钥。该临时会话密钥的值是一个由参与各方提供的、输入共同作用得到的、对于参与各方而言是相同的函数值。密钥协商技术使得通信的各方能够在不必频繁更换其基本密钥的情况下，得到类似于一次一密系统的安全性。

在5.2.2节中，介绍了椭圆曲线公钥密码体系中的密钥协商方案，在分析、研究现有的密钥协商方案的基础上，给出了作者所设计的XKAS密钥协商方案，并分析了该方案的运行性能和安全性。XKAS密钥协商方案基于椭圆曲线有限群上的椭圆曲线离散对数ECDLP问题的求解困难性，通过引入可信第三方权威认证机构，实现了简单实用的高效密钥协商。与之前的各种类似的密钥协商协议方案相比，XKAS方案操作简单、高效，能够应用于各种软硬件环境中。随后，还针对不同的应用需要，对XKAS密钥协商方案进行了扩展，得到了单方XKAS密钥协商方案和三方XKAS密钥协商方案。本节将在5.2.2节的研究基础上，给出XKAS密钥协商方案的具体实现方法。

由5.2.2节可知，对于XKAS方案，有如下约定。

设椭圆曲线$E(GF(q))$是定义在有限域$GF(q)$上的一条安全椭圆曲线，其上随机选取的基点为G，设$n=\#E(GF(q))$是椭圆曲线E的阶，r是n的大素数因子，则通信的双方A和B须事先按照算法5.1产生自己的私钥SK和公钥PK，并将所产生的公钥PK

置于可信的第三方认证中心CA 处。

以通信一方A 为例，只有两个通信参与方的点对点型XKAS 密钥协商方案见算法7.4。

对于通信方B 而言，其所执行的XKAS 密钥协商方案与算法7.4 相似，这里就不再介绍了。

算法7.4　XKAS 密钥协商方案

输入：B 的公钥PK_B，A 的私钥SK_A，收到的密钥协商报文S_B 输出：临时会话密钥K，密钥协商报文S_A
1. 随机选择一整数$k_A \in [1, r-1]$； 2. 计算$S_A = k_A PK_B$； 3. 将S_A发送给B； 4. 计算$K_{AB} = k_A SK_A S_B$； 5. 返回$K = H(K_{AB})$。

可以对标准的XKAS 密钥协商方案进行扩展和改进，以适应各种不同的应用需求。算法7.5 和算法7.6 描述了一方匿名的单方XKAS 密钥协商方案。这里，假设在通信的双方A 和B 中，B 为匿名方。设A 事先选取的私钥为SK_A，按照算法5.1 产生的公钥为PK_A被置于可信的第三方认证中心CA 处。椭圆曲线密码体系的系统参数同算法7.4。

算法7.5　匿名方B 执行的单方XKAS 密钥协商方案

输入：A 的公钥PK_A 输出：临时会话密钥K，密钥协商报文S
1. 随机选择一整数$k \in [1, r-1]$； 2. 计算$S = kPK_A$；[密钥协商报文]

3. 将S发送给A；
4. 计算$K=kG$；[临时会话密钥]
5. 返回K。

算法7.6 非匿名方A执行的单方XKAS密钥协商方案

输入：A的私钥SK_A，收到的密钥协商报文S
输出：临时会话密钥K

1. 计算$K=SK_A\times S$；
2. 返回K。

当通信主体不止两个时，需要对上述的XKAS密钥协商方案具体的实施过程进行适当的改进。现以三个通信主体参与通信时的情况为例，说明这一改进的具体实施方案。

设通信的三方为A、B和C，他们的密钥对均由算法5.1产生，他们的私钥分别为SK_A、SK_B和SK_C，公钥分别为PK_A、PK_B和PK_C，则对任一通信参与方(如通信方A)而言，三方XKAS密钥协商方案如算法7.7。

算法7.7 三方XKAS密钥协商方案

输入：C的公钥PK_C，A的私钥SK_A，由C发送的密钥协商报文Y_C和Z_C
输出：临时会话密钥K，密钥协商报文X_A和Z_A

1. 随机选择一整数$k_A\in[1,r-1]$；
2. 计算$X_A=k_APK_C$，将X_A发送给B；
3. 计算$Z_A=k_AZ_C$，将Z_A发送给B；

4. 计算 $K_{BCA}=k_A S K_A Y_C$；
5. 返回 $K=K_{BCA}$。

基于上述的XKAS密钥协商方案，得以在Windows 2000平台和Linux平台上分别实现。实验结果表明，这种密钥协商方案简单高效、易于实现，无需繁琐的身份鉴别认证过程，也不需要引入时戳服务器，只需要一次数据交换即可直接完成密钥协商任务，能够抵抗目前已知的各种攻击方案，安全性很高。与之前的各种类似的密钥协商方案相比，XKAS方案操作简单、高效，计算开销和空间需求很低，能够应用于如计算机、通信网络、智能卡、手机、移动通信等各种软硬件环境中，具有很好的应用前景。

7.3 XKDS密钥分配方案

密钥分配协议是密码密钥学所研究的核心问题之一，它主要研究如何在开放式网络通信的环境中，实现系统中不同成员之间的密钥安全传输等问题。对通信系统的安全具有极其重要的意义。它不仅影响着系统的安全性，而且还涉及系统的可靠性、有效性和经济性等。

与上文所介绍的密钥协商方案(Key Agreement Scheme)不同，密钥分配方案(Key Distribution Scheme)是这样一种机制：首先，由通信的一方选择一个秘密的通信密钥，然后将它通过某一种安全机制传输给通信的其他参与方，从而在开放式网络通信环境中，实现系统中不同成员之间的密钥安全传输，这种安全机制就是密钥分配方案。本节将在5.2.3节的研究基础上，研究XKDS密钥

分配方案的具体实现问题。

对于XKDS方案，系统参数约定如下。

对于随机选定大素数p，设椭圆曲线$E(GF(p)):y^2=x^3+ax+b \bmod p$是定义在有限域$GF(p)$上的一条安全椭圆曲线，其上随机选取的基点为$G$，设$n=\#E(GF(p))$是椭圆曲线$E$的阶，$r$是$n$的一个大素数因子。

设通信的双方为A和B，他们的密钥对均根据算法5.1产生。他们所选取的私钥分别为SK_A和SK_B，公钥分别为PK_A和PK_B，且约定由A负责产生临时会话密钥，并向通信方B分配该临时会话密钥，则算法7.8和算法7.9描述了点对点型XKDS密钥分配方案的操作步骤。

算法7.8　发送方A执行的XKDS密钥分配方案

输入：B的公钥PK_B
输出：临时会话密钥K，密钥分配报文S

1. 随机选择一整数$k\in[1,r-1]$；
2. 计算$R=k\times PK_B$；
3. 计算$S=Sig_A(R)$；[数字签名]
3. 将S发送给B；
4. 计算$K=kG$；[临时会话密钥]
5. 返回K。

算法7.9　接收方B执行的XKDS密钥分配方案

输入：B的私钥SK_B，收到的密钥分配报文S
输出：临时会话密钥K

1. 从S中析出R;
2. 检查方程$Ver(PK_A,R,S)=0$是否成立;
   2.1 若不成立,则报错并退出;
3. 计算$K=SK_B\times R$;
4. 返回K。

当通信主体不止两个时,5.2.3节给出了一种有多个通信参与方参与的会议主席制单播会议XKDS密钥分配扩展方案。该方案实际上是由多组点对点型XKDS密钥分配方案复合而成的。

现假设由通信主体A担任会议主席,负责产生和分发临时会话密钥。会议的其他各参与会员为$M_1,M_2,\cdots,M_n$,记为M_i。他们的密钥对均根据算法5.1产生。现设会议主席的密钥对为(PK_A,SK_A),各会员的密钥对为(PK_i,SK_i),则算法7.10描述了会议主席产生和分发临时会话密钥的工作过程,其他参与成员则依据算法7.11解读所收到的密钥分配报文S_i。

算法7.10 会议主席A执行的XKDS密钥分配方案

输入:会议其他各参与方$M_1,M_2,\cdots,M_n$的公钥PK_i
输出:临时会话密钥K,密钥分配报文S_i

1. 随机选择一整数$k\in[1,r-1]$;
2. For i from 1 to n do
   2.1 计算$R_i=k\times PK_i$;
   2.2 计算$S_i=Sig_A(R_i)$;[数字签名]
   2.3 将S_i发送给M_i;
3. 计算$K=kG$;[临时会话密钥]
4. 返回K。

算法7.11　接收方M_i执行的XKDS密钥分配方案

输入：M_i的私钥SK_i，收到的密钥分配报文S_i 输出：临时会话密钥K
1. 从S_i中析出R_i； 2. 检查方程$Ver(PK_A, R_i, S_i)=0$是否成立； 2.1　若不成立，则报错并退出； 3. 计算$K=SK_i \times R_i$；[临时会话密钥] 4. 返回K。

基于上述的XKDS密钥分配方案，在Windows 2000平台和Linux平台上分别得以实现。实现结果表明，XKDS密钥分配方案简单高效、安全可靠、易于实现。

7.4　数据加密算法

本书的第5.3节介绍了椭圆曲线公钥密码体系中数据加密算法，介绍了作者所提出的基于密钥共享机制的混合密码加密算法XHES(Xiao's Hybird Encryption Scheme)和改进的ElGamal椭圆曲线加密算法X-ElGamal。本节将在第5.3节的研究基础上，给出两种加密算法的具体实现。

这里，加密系统参数约定如下。

对于随机选定大素数p，设椭圆曲线$E(GF(p)): y^2=x^3+ax+b \bmod p$是定义在有限域$GF(p)$上的一条安全椭圆曲线，其上随机选取的基点为$G$，设$n=\#E(GF(p))$是椭圆曲线$E$的阶，$r$是$n$的一个大素数因子。

设通信主体 A 的基本密钥对是根据算法 5.1 产生的，设其私钥为 SK_A，其公钥 PK_A 则被置于可信的第三方认证中心 CA 处，并假定由通信方 B 对明文消息 M 进行加密。

1. X-ElGamal 加密算法

X-ElGamal 椭圆曲线公钥加密算法是对ElGamal 型椭圆曲线公钥加密算法的改进，该算法无需求逆元素的计算，运行速度优于其他椭圆曲线加密算法，而且充分利用了屏蔽参数 $k\times G$ 的两个坐标分量，具有较好的时间和空间性能。

算法 7.12 和算法 7.13 描述了 X-ElGamal 算法的加密过程和解密过程。

算法 7.12　X-ElGamal 算法的加密过程

输入：A 的公钥 PK_A，待加密的明文消息块 M
输出：密文数据块 c

1. 将 M 表示成 $GF(p)$ 上的一对元素 $m=(m_1,m_2)$；
2. 随机选择整数 $k\in[1,r-1]$；
3. 计算 $c_0=k\times PK_A$；
4. 计算 $Q=k\times G=(y_1,y_2)$；
5. 计算 $E(m_1,m_2,k)=(c_0,y_1+m_1,y_2+m_2)=(c_0,c_1,c_2)$；
6. 返回密文数据块 $c=(c_0,c_1,c_2)$。

算法 7.13　X-ElGamal 算法的解密过程

输入：A 的私钥 SK_A，待解密的密文数据块 c
输出：明文消息块 M

1. 计算$Q=SK_A\times c_0=(y_1,y_2)$；
2. 计算$D(c_0,c_1,c_2)=(c_0,c_1-y_1,c_2-y_2)=(m_1,m_2)$；
3. 合并元素$m=(m_1,m_2)$，得明文消息块M；
4. 返回M。

在Windows 2000平台和Linux平台上分别对上述算法进行了实现。实验结果表明，在Intel Pentium III 700的计算机上，X-ElGamal算法的加密速度可以达到0.008 MB/s，与其他椭圆曲线加密算法相比，具有较高的加密效率和性能，但其运行性能仍远远低于同等安全性的对称密码加密算法的加密性能，不能满足实际应用的需要。因此，作者进一步提出了基于密钥共享体系的混合密码加密算法XHES(Xiao's Hybird Encryption Scheme)。

2. XHES混合密码加密算法

XHES算法的特点在于它兼有两类加密体系的优点，不仅具有加密速度快、强度高等优点，而且具有便捷的密钥分发和管理上的独到优势。它由密钥共享协议和对称密码算法两大部分组成，此外，通常还需要一个单向密钥变换算法用于完成不同形式密钥的转换任务。

在设计XHES混合密码加密算法的具体实施时，有如下一些基本设计考虑。

① 实际应用中，经常需要加密算法除了具备信息保密的基本功能以外，还需要具备身份鉴别和认证等附加功能。因此，选择具备身份鉴别和认证功能的点对点型XKDS密钥分配方案作为XHES混合密码加密算法中的密钥共享协议。

② 从安全性和信息保密角度出发，选择高级加密标准AES作

为XHES混合密码加密体系中的对称密码算法。

③ 由XKDS密钥分配方案分配的临时会话密钥实际上是安全椭圆曲线$E(GF(p))$上的一个点，而AES算法所使用的密钥实际上是一个指定长度的字符串。所以还需要一个单向密钥变换算法来完成两种形式密钥的转换任务，即将椭圆曲线上一个点转换为一个指定长度的密钥字符串。因此，选择安全摘要算法SHA来完成这一任务。

作为一个简单的实施实例，依据上述三点考虑，有算法7.14和算法7.15。

算法7.14　XHES算法的加密过程

输入：A的公钥PK_A，B的私钥SK_B，待加密的明文消息M 输出：密文数据报文C
1. 随机选择整数$k\in[1,r-1]$； 2. 计算$R=k\times PK_A$； 3. 计算$S=Sig_B(R)$； 4. 计算$K=SHA(k\times G)$；[加密密钥] 5. 计算$E=AES_Enc(K,M)$； 6. 返回密文数据报文$C=(S,E)$。

算法7.15　XHES算法的解密过程

输入：A的私钥SK_A，B的公钥PK_B，待解密的密文数据报C 输出：明文消息M
1. 从C中析出S和E； 2. 从S中析出R；

3. 检查方程$Ver(PK_B,R,S)=0$是否成立；
   3.1 若不成立，则报错并退出；
4. 计算$K=SHA(SK_A\times R)$；[解密密钥]
5. 计算$M=AES_Dec(K,C)$；
6. 返回明文消息M。

为比较算法性能，作者在Windows 2000平台和Linux平台上分别对算法7.14和算法7.15进行了实现。实验结果表明，在Intel Pentium III 700的计算机上，当源数据文件大小为2 MB以上时，XHES算法的加密速度可以达到3.6 MB/s。这一速度远远高于直接使用椭圆曲线公钥密码加密的速度，已达到和接近同等安全性条件下的对称密码加密算法的加密性能，足以满足实际应用的需要。它不仅具有加密速度快、强度高等优点，而且具有便捷的密钥分发和管理上的独特优势。XHES混合密码加密方案兼有两类加密体系的优点：安全可靠、简单高效、能够应用于各种软硬件环境中，具有很好的使用价值。

7.5　XECDS数字签名方案

第5.4节详细介绍了数字签名技术的原理和方法，研究了椭圆曲线公钥密码体系中的数字签名方案，在分析现有的基于离散对数问题的ElGamal类数字签名方案的基础上，5.4.1节以一个统一的形式描述了所有这类基于椭圆曲线离散对数问题的普通数字签名方案XECDS，并给出了具体的设计实例——XECDS-I签名方案。5.4.3节在XECDS数字签名方案的研究基础上，介绍了盲数字签名方案工作过程，设计了一个基于XECDS-I签名方案的

盲数字签名方案XECBDS。5.4.4 节则介绍了代理数字签名方案的特点和工作流程，给出了一个基于XECDS-I 签名方案的代理签名方案XECPDS，并对其进行了详细的分析。而5.4.5 节则针对现有代理签名方案的不足，提出并设计了一个具有限制代理人代理权限的受控代理数字签名方案 XECLPDS，并证明了该方案的正确性，分析了该方案的特点。本节将在第5.4 节的理论研究的基础上，介绍 XECDS-I 普通签名方案、XECBDS 盲数字签名方案和 XECPDS 代理数字签名方案的实现算法。

本节中系统参数约定与前文类似。

1. XECDS-I 普通签名方案

XECDS-I 普通签名方案是XECDS 系列普通数字签名方案的一个具体实施实例，其签名方程和验证方程如式(5.5)和式(5.6)所示，所选用的消息摘要算法为安全摘要算法SHA-1。

算法7.16 和算法7.17 分别描述了XECDS-I 普通签名方案对数据报文 m 进行数字签名和验证的签名过程和验证过程，并假定由通信方 A 对明文消息 M 进行数字签名。

算法7.16 XECDS-I 数字签名算法的签名过程

输入：A 的私钥 SK_A，待签名的数据报文 m 输出：数字签名 S
1. 随机选择整数 $k\in[1,r-1]$； 2. 计算 $Q=k\times G$； 3. 计算 $H(m)=SHA(m)$； 4. 计算 $s=H(m)\times k+Q_x\times SK^{-1}$； 5. 返回数字签名 $S=(Q,s)$。

算法7.17　XECDS-I数字签名算法的验证过程

输入：A的公钥PK_A，待核实的数据报文m和待验证的数字签名S 输出：签名真伪标志$DSignIsTrue$
1. 计算$H(m)=SHA(m)$； 2. 计算$R=H(m)\times Q$； 3. 计算$T=s\times G-Q_x\times PK_A$； 4. 置$DSignIsTrue=(R==T)$； 5. 返回数字签名真伪标志$DSignIsTrue$。

当签名真伪标志$DSignIsTrue$为真时，说明待核实的数据报文m和待验证的数字签名S是一致的；否则，说明待核实的数据报文m和待验证的数字签名S是不一致的：或者待核实的数据报文m被篡改；或者待验证的数字签名S被改动；或者两者均被改动过。

2. XECBDS盲数字签名方案

XECDS-I签名方案只能生成适用于普通应用需求条件的数字签名。对于某些复杂特殊条件下的应用需求，XECDS-I签名方案就不能适用了。下面将要介绍的是5.4.3节讨论的椭圆曲线公钥密码体系下XECBDS盲数字签名方案的具体实施方案。

这里假定由用户A委托签名者B对明文消息M进行盲数字签名，所选消息摘要算法为SHA-1算法，则下面的算法7.18～算法7.22描述了图5-11所示的XECBDS盲数字签名方案中的各主要关键算法。

算法 7.18 XECBDS 盲数字签名方案:参数初始化算法

输入:椭圆曲线密码体系基本参数 输出:临时参量 k,K 说明:由签名者 B 执行
1. 随机选择整数 $k\in[1,r-1]$; 2. 计算 $K=k\times G$; 3. 返回 k 和 K,并将 K 发送给用户 A。

算法 7.19 XECBDS 盲数字签名方案:盲变换算法

输入:待签名的明文消息 M,临时参量 K 输出:盲消息 m 说明:由用户 A 执行
1. 验证 K 是否在椭圆曲线 E 上; 2. 计算 $h=SHA(M)$; 3. 随机选择整数 $\alpha,\beta\in[1,r-1]$; 4. 计算 $R=\alpha\times K+\beta\times G$; 5. 计算 $m=\alpha K_x R_x^{-1}\times h$;[盲消息] 6. 返回盲消息 m,并发送给签名者 B。

算法 7.20 XECBDS 盲数字签名方案:数字签名算法

输入:待签名的盲消息 m,临时参量 k 和 K,签名者 B 的私钥 SK 输出:盲消息 m 的数字签名 $\tilde{s}$ 说明:由签名者 B 执行
1. 计算 $\tilde{s}=k\times m+K_x\times SK^{-1}$; 2. 返回盲消息 m 的数字签名 $\tilde{s}$,并发送给用户 A。

算法7.21　XECBDS盲数字签名方案：逆盲变换算法

输入：盲消息m的数字签名$\tilde{s}$，临时参量R和K 输出：明文消息M的盲数字签名S 说明：由用户A执行
1. 计算$s=\tilde{s}\times R_x\times K_x^{-1}+\beta\times h$； 2. 计算$t=R_x \bmod r$； 3. 返回盲数字签名$S=(t,s)$。

算法7.22　XECBDS盲数字签名方案：验证算法

输入：B的公钥PK，待核实的数据报文M和待验证的数字签名S 输出：签名真伪标志$DSignIsTrue$ 说明：由消息的验证者执行
1. 析出t和s； 2. 计算$h=SHA(M)$； 3. 计算$T=h^{-1}\times(s\times G-t\times PK)$； 4. 置$DSignIsTrue=(t==T_x \bmod r)$； 5. 返回数字签名真伪标志$DSignIsTrue$。

3. XECPDS代理数字签名方案

为了解决数字签名权力的委托代理转移问题，人们提出了代理数字签名方案。下面将要介绍的是5.4.4节所讨论的椭圆曲线公钥密码体系下的XECPDS代理数字签名方案的具体实施方案。

这里假定由代理签名者B根据原始签名者A的委托，对明文

消息 M 进行代理数字签名，所选消息摘要算法为 SHA-1 算法，则下面的算法 7.23～算法 7.26 描述了 5.4.4 节所阐述的 XECPDS 代理数字签名方案中的主要关键算法。

算法 7.23 XECPDS 代理数字签名方案：生成委托信息

输入：椭圆曲线密码体系基本参数，原始签名者 A 的私钥 SK_A 输出：委托信息 SK_σ，Q 说明：由原始签名者 A 执行
1. 随机选择整数 $k\in[1,r-1]$； 2. 计算 $Q=k\times G$； 3. 计算 $SK_\sigma^{-1}=SK_A^{-1}+k\times Q_x$； 4. 返回 SK_σ，Q，并将(SK_σ，Q)发送给代理签名者 B。

算法 7.24 XECPDS 代理数字签名方案：接受委托请求

输入：委托信息(SK_σ，Q)，代理签名者 B 的私钥 SK_B 和公钥 PK_B 输出：代理签名私钥 SK_σ 说明：由代理签名者 B 执行
1. 验证等式 $SK_\sigma^{-1}\times G=PK_A+Q_x\times Q$ 是否成立； 2. 计算 $SK_\sigma^{-1}=SK_\sigma^{-1}+SK_B^{-1}\times(PK_B)_x$； 3. 返回代理签名私钥 SK_σ。

算法 7.25 XECPDS 代理数字签名方案：数字签名算法

输入：代理签名私钥 SK_σ，待签名的明文消息 M，委托信息 Q 输出：代理数字签名 S 说明：由代理签名者 B 执行，算法同 XECDS-I 数字签名算法

1. 随机选择整数$k\in[1,r-1]$；
2. 计算$K=k\times G$；
3. 计算$H=SHA(m)$；
4. 计算$s=H\times k+K_x\times SK_a^{-1}$；
5. 返回代理数字签名$S=(K,s,Q)$。

算法7.26　XECPDS代理数字签名方案：签名验证算法

输入：原始签名者A的PK_A，代理签名者B的公钥PK_B，待核实的明文消息M和待验证的代理数字签名S
输出：签名真伪标志$DSignIsTrue$
说明：由消息的接收验证者执行

1. 计算代理签名公钥$PK_a=PK_A+Q_x\times Q+(PK_B)_x\times PK_B$；
2. 计算$H=SHA(m)$；
3. 计算$R=H\times K$；
4. 计算$T=s\times G-K_x\times PK_a$；
5. 若$R=T$，则置$DsignIsTrue$为真，否则置为假。

4. XECLPDS受控代理数字签名方案

针对代理签名权力的可控性问题，5.4.5节提出了XECLPDS受控代理数字签名方案，这里将介绍椭圆曲线公钥密码体系下的XECLPDS受控代理数字签名方案的具体实施方案。

这里假定由代理签名者B根据原始签名者A的委托，对明文消息M进行代理数字签名，所选消息摘要算法为SHA-1算法，则下面的算法7.27～算法7.30描述了5.4.5节和图5-12所阐述的XECLPDS受控代理数字签名方案中的各主要的关键算法。

算法 7.27　XECLPDS 代理数字签名方案：生成委托请求

输入：椭圆曲线密码体系基本参数，原始签名者 A 的私钥 SK_A
输出：委托请求 (C_p, M_p)
说明：由原始签名者 A 执行

1. 生成授权证书 A_p，签署得授权证书 C_p；
2. 随机选择整数 $k \in [1, r-1]$；
3. 计算委托参数 $Q_p = k \times G$；
4. 计算 $H_p = SHA(C_p, Q_p)$；
5. 计算授权参数 $S_p = H_p \times SK_A + k$；
6. 生成委托授权信息 $M_p = (S_p, Q_p)$；
7. 返回委托请求 (C_p, M_p)，并将 (C_p, M_p) 发送给代理签名者 B。

算法 7.28　XECLPDS 代理数字签名方案：接受委托请求

输入：委托请求 (C_p, M_p)，原始签名者的公钥 PK_A，代理签名者 B 的私钥 SK_B
输出：代理签名私钥 SK_p
说明：由代理签名者 B 执行

1. 验证授权证书 C_p 的合法性以及授权文书 A_p 的合理性；
2. 从委托授权信息 M_p 中析出委托参数 Q_p 和授权参数 S_p；
3. 计算 $H_p = SHA(C_p, Q_p)$；
4. 验证等式 $S_p \times G = Q_p + H_p \times PK_A$ 是否成立；
5. 计算代理签名私钥 $SK_p = S_p + SK_B \times H_p$；
6. 返回代理签名私钥 SK_p。

算法7.29　XECLPDS代理数字签名方案:数字签名算法

输入:代理签名私钥SK_p,待签名的明文消息M,委托授权信息M_p
输出:代理数字签名S
说明:由代理签名者B执行

1. 按约定的普通数字签名算法,生成普通数字签名$S'=Sig(SK_p,m)$;
2. 生成的代理数字签名$S=(S',Q_p)$;
3. 请求认证,获取时戳证书T_p;
4. 返回完整的数据电文$M=(m,S,C_p,T_p)$。

算法7.30　XECLPDS代理数字签名方案:签名验证算法

输入:原始签名者A的PK_A,代理签名者B的公钥PK_B,待核实的数据电文M
输出:签名真伪标志$DSignIsTrue$
说明:由消息的接收验证者执行

1. 从M中析出消息m、代理数字签名S、授权证书C_p和时戳证书T_p;
2. 验证授权证书C_p和时戳证书T_p以及代理签名行为的合法性;
3. 从代理数字签名S中析出普通数字签名参数S'和委托参数Q_p;
4. 计算$H_p=SHA(C_p,Q_p)$;
5. 计算代理签名公钥$PK_p=Q_p+H_p\times(PK_A+PK_B)$;

6. 按约定的普通数字签名算法，验证方程 $Ver(S', PK_p, m) =$ True 是否成立；
7. 若方程成立，则置 *DsignIsTrue* 为真，否则置为假。

作者在 Windows 2000 平台和 Linux 平台上分别基于上述算法，对 XECDS-I 普通签名方案、XECBDS 盲数字签名方案、XECPDS 代理数字签名方案和 XECLPDS 受控代理数字签名方案进行了具体的实现，完成了四套简单实用、功能齐全、分别基于 XECDS-I 普通签名方案、XECBDS 盲数字签名方案、XECPDS 代理数字签名方案和 XECLPDS 受控代理数字签名方案的应用系统。

参考文献

[1] 肖攸安. 网络信息安全中的椭圆曲线公钥密码体系的研究[D]. 武汉:武汉理工大学信息工程学院,2003.

[2] 卢开澄. 计算机密码学——计算机网络中的数据保密与安全[M]. 第二版. 北京:清华大学出版社,1998.

[3] 冯登国,裴定一. 密码学导引[M]. 北京:科学出版社,1999.

[4] 肖攸安,周祖德. 超椭圆曲线可控代理签名方案的研究[J]. 计算机工程与应用. 2006,42(4):20-23.

[5] Symantec Corp. Symantec Internet Security Threat Report Volume VIII [R/OL], (2005-09-20). http://www.symantec.com.

[6] 水清木华研究中心. 2005 年中国电子商务盈利模式研究报告[R]. 北京:北京水清木华科技有限公司,2005. 07.

[7] 侯云智. 群论基础教程[M]. 山东:山东大学出版社,1997.

[8] Blake, I., Seroussi, G., and Smart, N. Elliptic Curves in Cryptography [M], Cambridge: Cambridge University Press,1999.

[9] Bailey, D. V., Paar, C. Efficient Arithmetic in Finite Fields Extensions with Applica-tion in Elliptic Curve Cryptography [J], Journal of Cryptology, 2002(6).

[10] 白国强. 椭圆曲线密码及其算法研究[D]. 西安:西安电子科技大学通信工程学院,2000.

[11] Fastenberg, Lisa A., Uniform Bounds for Integral Points on

Families of Elliptic Curves[J], Journal of Number Theory, 2002, 92(1): 197-203.

[12] 肖攸安, 李腊元. 数字签名技术的研究[J]. 武汉理工大学学报(交通科学与工程版). 2002, 26(6): 737-740

[13] Menezes, A., van Oorschot, P., Vanstone, S. Handbook of Applied Cryptography[M], Florida: CRC Press, 1996.

[14] Schoof, R., Elliptic Curves over Finite Fields and the Computation of Square Roots mod p[J], Mathematics of Computing, 1985, 44: 483-494.

[15] 李学俊, 敬忠良, 戴冠中等. 基于椭圆曲线离散对数问题的公钥密码[J]. 计算机工程与应用. 2002, 38(6): 20-22.

[16] 王许书, 王昭顺, 曲英杰. 基于复合域上的椭圆曲线密码体系的计算算法[J]. 小型微型计算机系统. 2002, 23(8): 1007-1009.

[17] 王育民, 刘建伟. 通信网的安全——理论与技术[M]. 西安: 西安电子科技大学出版社, 1999.

[18] 张方国, 王常杰, 王育民. 基于椭圆曲线的数字签名与盲签名[J]. 通信学报. 2001, 22(8): 22-28.

[General Information]
书名=椭圆曲线密码体系研究
作者=肖攸安著
页数=248
SS号=11872724
DX号=
出版日期=2006.10
出版社=华中科技大学出版社